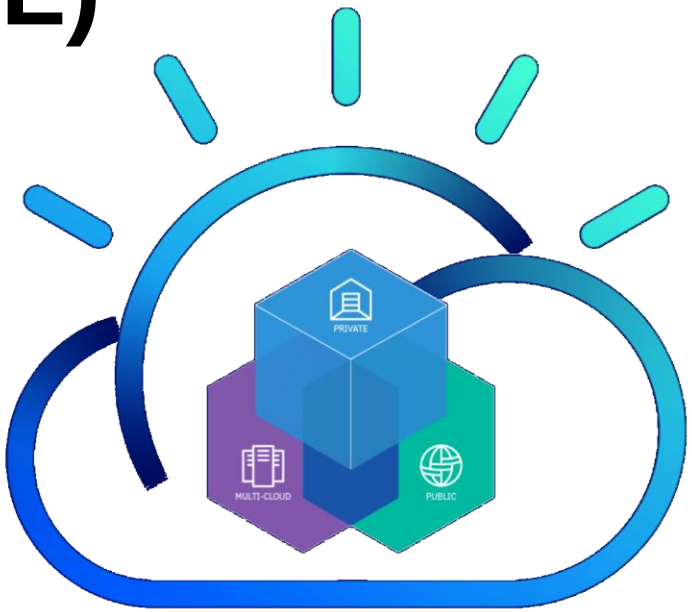
Build a Machine Learning System using IBM Cloud Private for Data Platform (ICP4D), Watson Studio Local (WSL) and Hadoop



Vish Kamat – vkamat@us.ibm.com IBM Data Science Expert Lab
Linda Liu – linda.liu@us.ibm.com IBM Data Science Expert Lab

Feb. 15, 2019

Please note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Agenda

- The Challenge and the Solution
- What is IBM Watson Studio Local (WSL)
- What is IBM Cloud Private for Data (ICP4D)
- ICP4D & Watson Studio Local Lifecycle Overview
- ICP4D (or WSL) and Hadoop
- QAs

Build a ML System using ICP4D (or WSL) and Hadoop

Challenge

You have a huge investment in Hadoop infrastructure but the tools in the ecosystem are not intuitive for business analyst and users. Most of your data is probably available within your HDFS along with your huge compute capacity but access to it is difficult.

Solution

Build machine learning and analytics on ICP4D platform or Watson Studio that provides variety of open source libraries and frameworks such as Spark, R, Keras, TensorFlow, Scikit-learn as a service within a kubernetes based platform behind your firewall. Along with cloud agility the platform provides end-end data management and analytics capability with collaboration and lifecycle management functions.

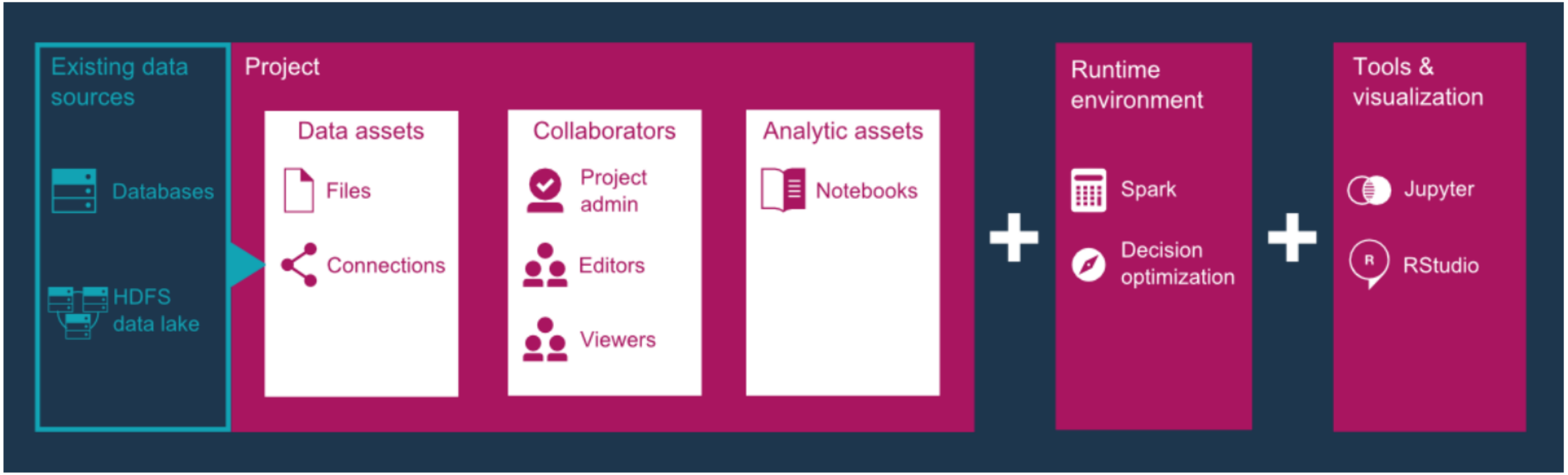
Hadoop connector provides seamless access to compute power available along with easy access to hdfs data.

Benefit

Accelerate your machine learning deployments in hours as oppose to months. Build quick to market business decision making Artificial Intelligence systems easily and quickly.

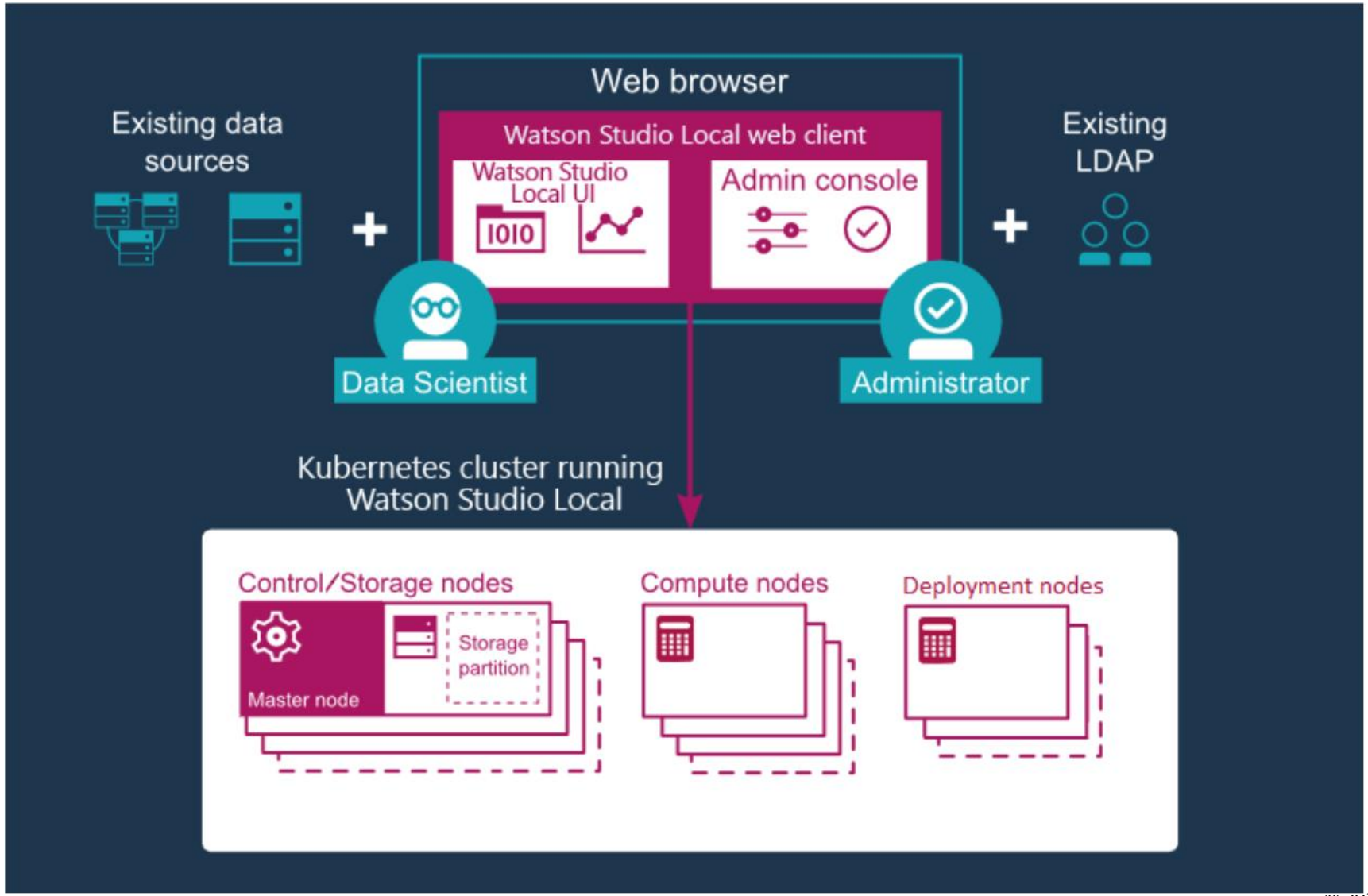
What is IBM Watson Studio Local (WSL)

- IBM Watson Studio Local is an out-of-the-box enterprise solution for data scientists and data engineers.
- It offers a suite of data science tools, such as RStudio, Spark, Jupyter, and Zeppelin notebooks, that are integrated with proprietary IBM technologies.



What is IBM Watson Studio Local (WSL)

Architecture



Why & What is IBM Cloud Private for Data (ICP4D)

ICP For Data Platform Functionalities

COLLECT

Data storage for all types of data with a consistent view across all deployments.

ORGANIZE

Integrate & govern your data to drive insights while mitigating compliance risks.

ANALYZE

Descriptive, predictive, prescriptive: to understand the current, predict the future and change outcomes.

IBM Cloud Private for Data – End-to-end AI workflow

Connect and discover content from multiple data sources in the cloud or on premises. Bring **structured** and **unstructured** data to one toolkit.

Find data (structured, unstructured) and AI assets (e.g., ML/DL models, notebooks, Watson Data Kits) in the **Project Assets** with intelligent search and giving the right access to the right users.

Clean and prepare your data with **Data Stage or ETL Tooling.** Use popular open source libraries to prepare unstructured data.

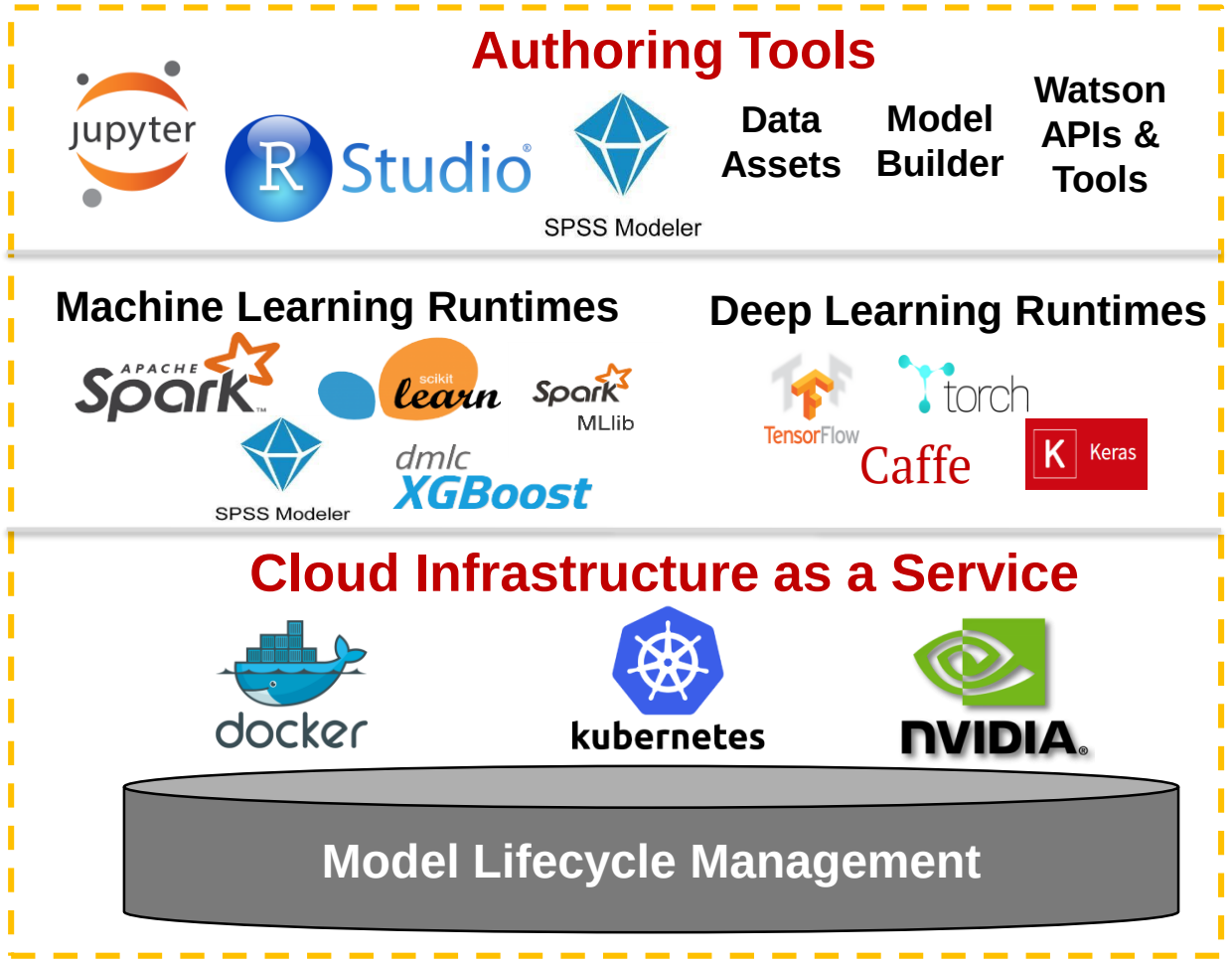
Democratize the creation of ML and DL models. Design your AI models **programmatically** or **visually** with the most popular **open source** and IBM ML/DL frameworks or leverage transfer learning on **pre-trained** models using **analytics tools** to adapt to your business domain. Train at scale on **GPUs** and

Deploy your models easily and have them **scale automatically** for online, batch or streaming use cases

Monitor the performance of the models in production and trigger automatic retraining and redeployment of models. Build **Enterprise Trust** with Bias Detection, Mitigation Model **Robustness** and Testing Service Model **Security**.

ICP For Data - Comprehensive set of tools for the end-to-end AI workflow

- Create, collaborate, deploy, and monitor
- 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
- Fit for Purpose Deployment
- Dynamic Resizing



Why & What is IBM Cloud Private for Data (ICP4D)

Extensible, Open API Platform

App Developers 

Data Engineers 

Business Partners 

Personalized, Collaborative Team Platform

Data Scientists 

Business Analysts & CxOs 

Data Stewards 

Cloud-native Scalable Data Micro Services

Collect

- ✓ Databases on-demand
- ✓ Data warehousing
- ✓ Fast Data Ingest
- ✓ Federated query

Organize

- ✓ Data integration
- ✓ Data curation
- ✓ Governance & policies
- ✓ Data Asset lifecycle management

Analyze

- ✓ Data Science
- ✓ Predictive Modeling & ML
- ✓ Data Visualization & Exploration
- ✓ Dashboards & reporting

Enterprise Data Catalog

Integrated Admin & Ops Dashboards, storage management, identity access management

IBM Cloud Private

Kubernetes

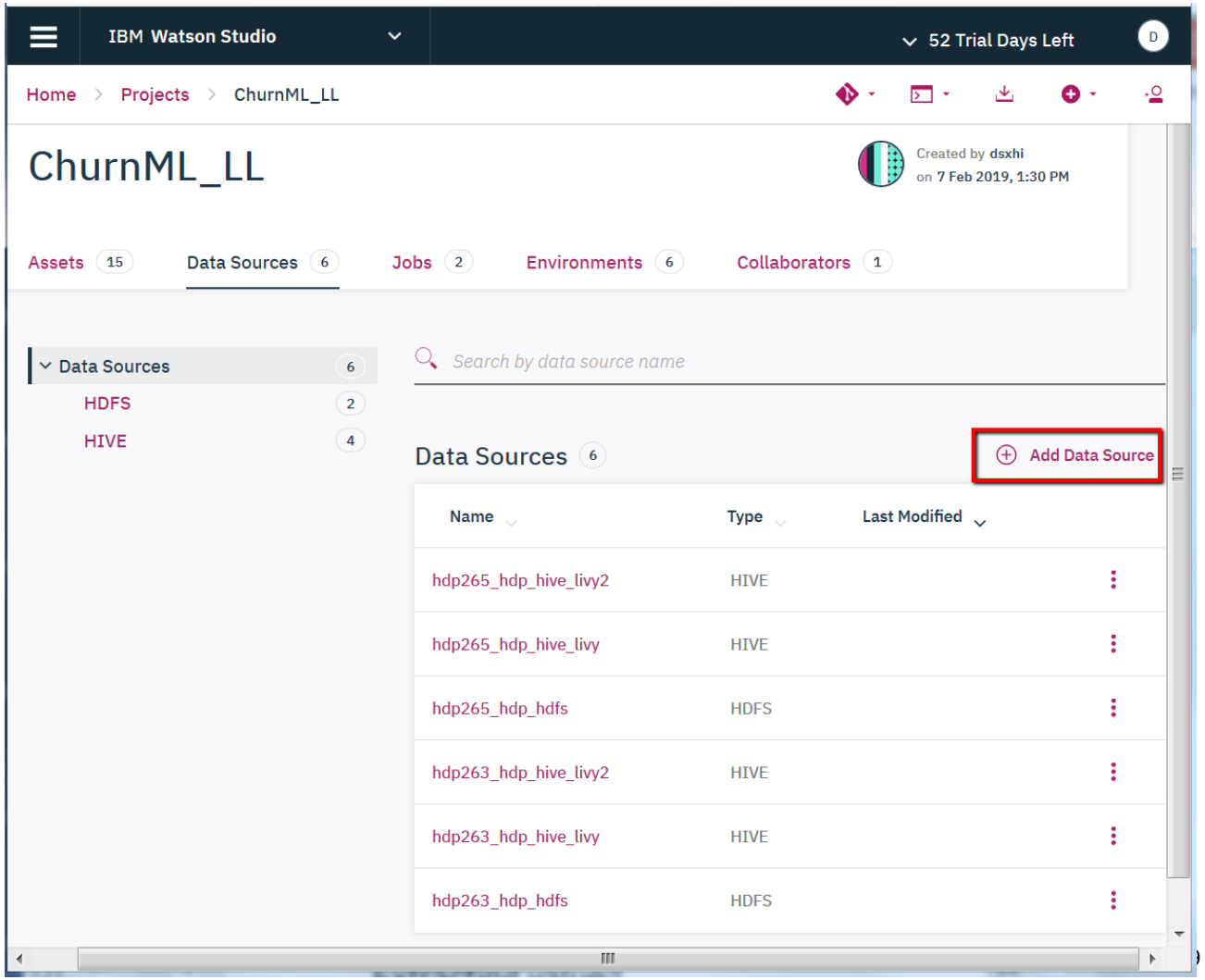
ICP4D/WSL – Lifecycle Overview

1. Create Data Sources, Load

Data Sets

You can create data sources and remote data sets for the following databases and services

- Big SQL
- DB2 Warehouse on Cloud
- DB2 for z/OS
- Hive
- HDFS
- Hyperledger Composer
- Informix
- Microsoft SQL Server (MSSQL)
- Netezza
- Oracle



ICP4D/WSL – Lifecycle Overview

2. Cataloging and Governance Data

a) Create/Import a data dictionary

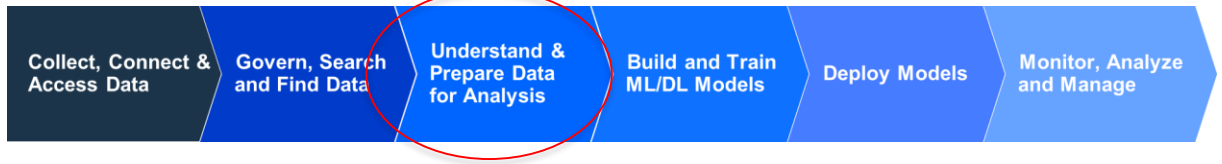
b) Creating terms and categories

c) Creating governance policies and rules

d) Governing analytical assets

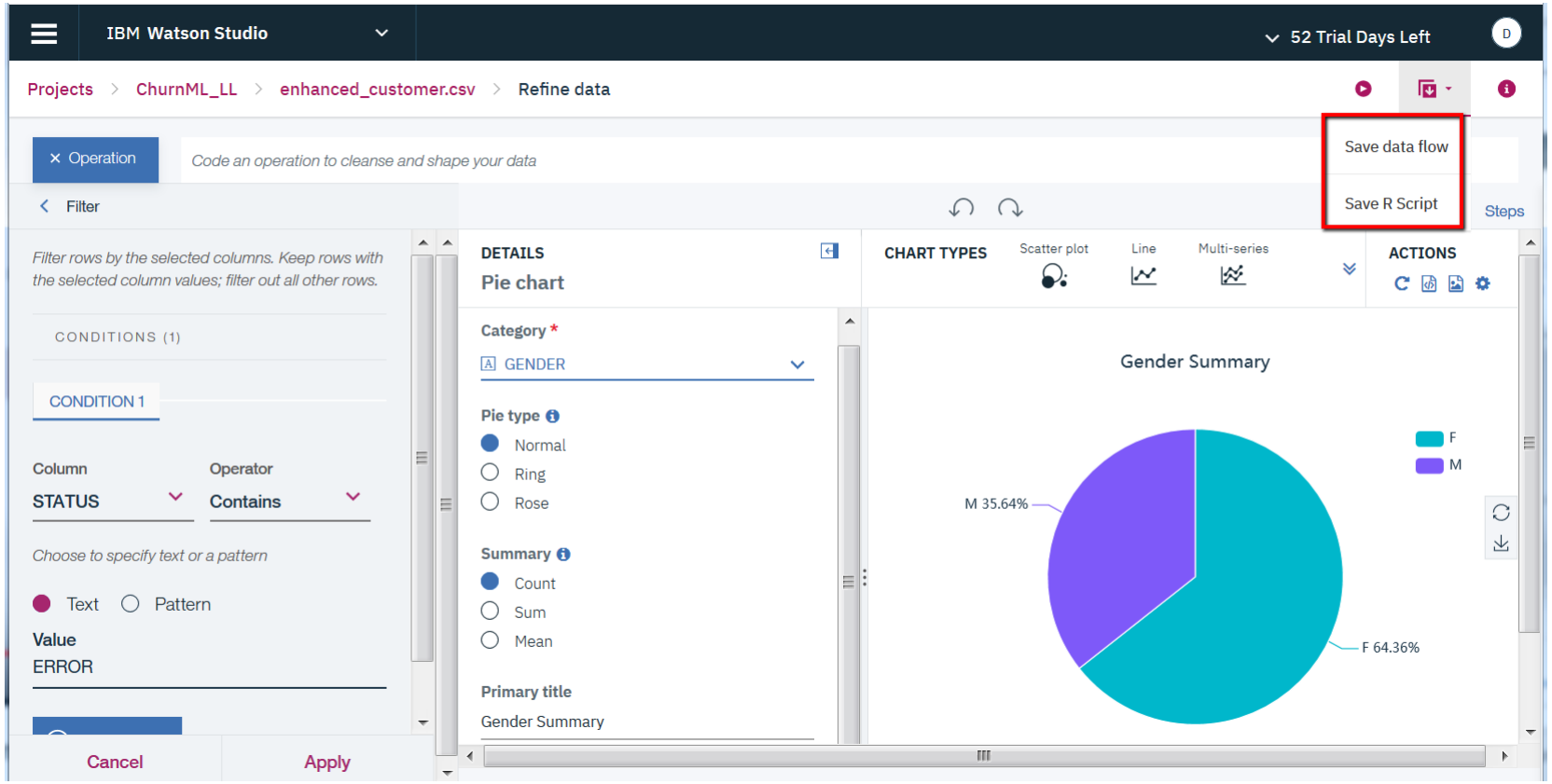
ICP4D/WSL – Lifecycle Overview

3. Refine Data Sets



Before you analyze local data sets, you can refine the data by cleansing and shaping it. Use the Data Refiner to perform the following tasks:

- Set up data
- Create a data flow
- Profile data
- Visualize data
- Run the flow as a job



ICP4D/WSL – Lifecycle Overview

4. Create Machine Learning Models

ICP4D/WSL Client provides tools to help you create and train machine learning models that can analyze data assets and extract value from them. Users can also deploy their models to make them available to a wider audience.

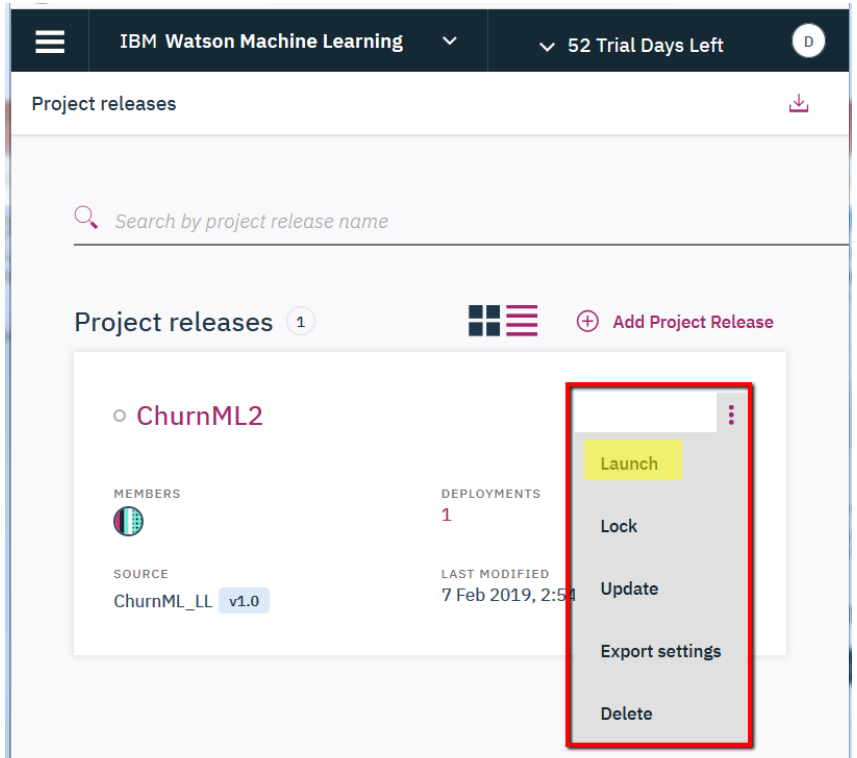
WSL supports the following machine learning model types:

- Spark ML
- PMML with online scoring
- Custom models with batch scoring
- scikit-learn
- XGBoost
- Keras
- TensorFlow
- WML

ICP4D/WSL – Lifecycle Overview

5. Model Management & Deployment

- To expose a checkpoint of assets to outside users, a Deployment Admin can create a project release and deploy the assets within it.
- A project release represents a project tag that can be launched as a production environment.



ICP4D/WSL – Lifecycle Overview

5. Model Management & Deployment

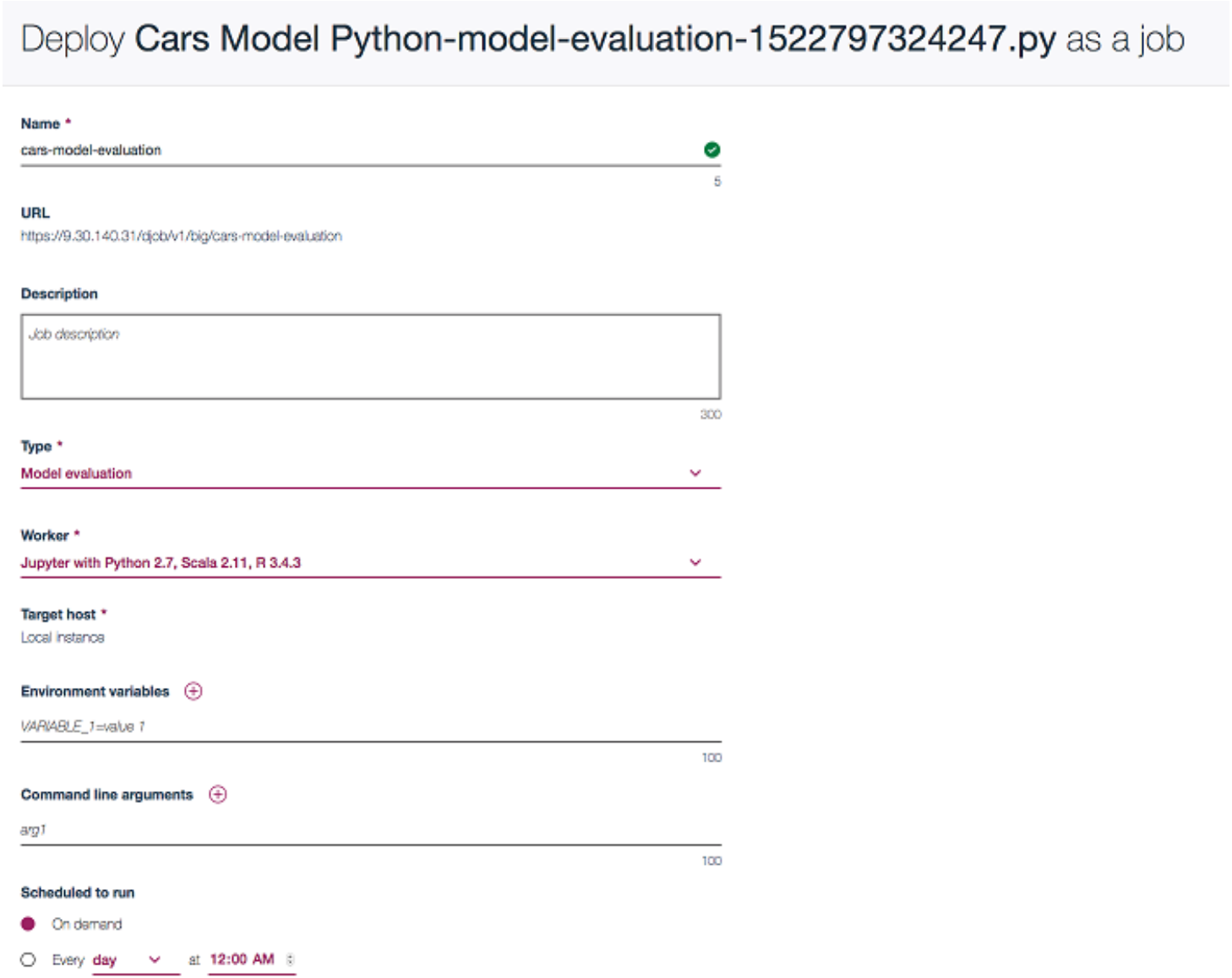
- An Admin can deploy an asset into a project release
- The deployment action creates a read-only snapshot of the asset
- The following types of assets can be deployed:

Asset	Job	Web service	App	Notes
Notebooks	Y	N	Y	Only Jupyter notebooks can be deployed. Zeppelin and H2O Flows are not supported by Watson Machine Learning.
Models	N	Y	N	Requires that you associate a batch scoring script with it. You cannot deploy Custom Batch models as a web service (must be Custom Online).
R Shiny	N	N	Y	Deploys the R code inside of an R pod.
Flows	Y	N	N	Flows from SPSS Modeler.
Decision Optimization Models	N	Y	N	Provides a REST API to submit and execute optimization jobs.
Scripts	Y	Y	N	
Model groups	N	Y*	N	* Can be deployed as a web service group.

ICP4D/WSL – Lifecycle Overview

6. Run Deployed Job Example

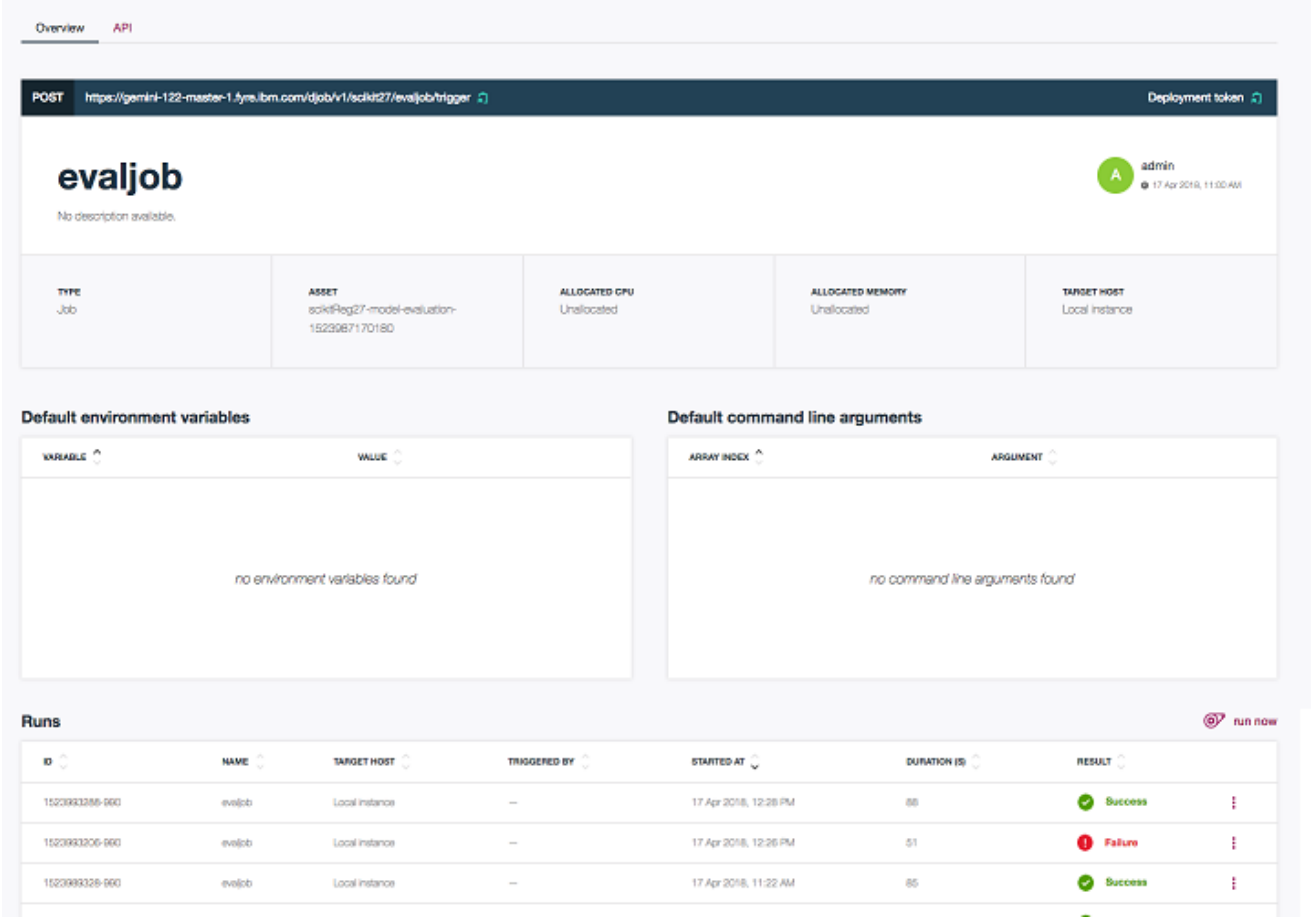
- You can navigate to the Assets tab in the Project release details page
- When a script is selected in the left panel of the assets table, the right panel of the assets table provides the button to create a job, which brings you to the Create job deployment page.



ICP4D/WSL – Lifecycle Overview

6a. Run Deployed Job Example – Overview Tab

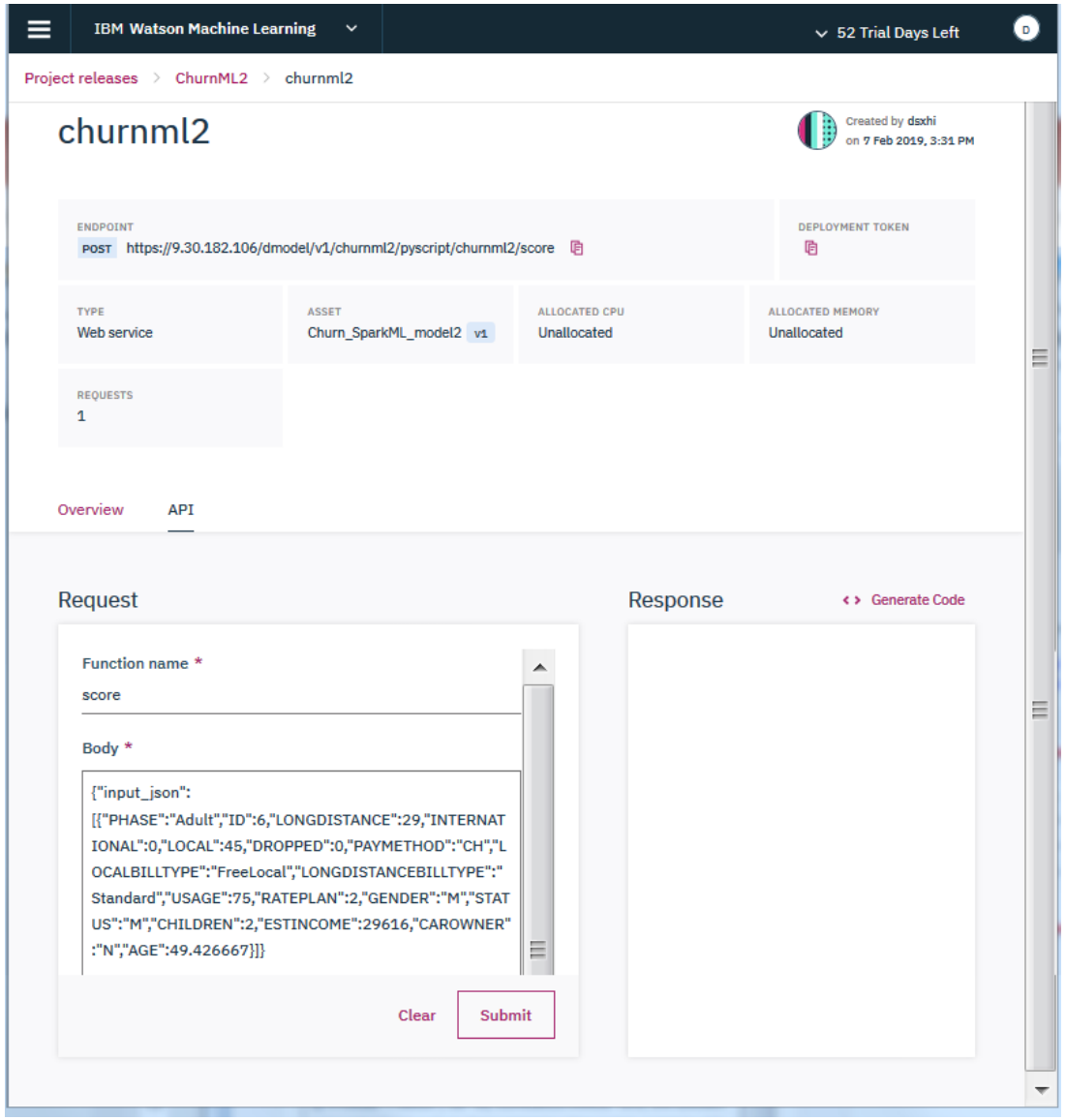
- A table of environments and arguments for the current job
- A table of all the runs for that particular job
- A button run now that triggers a job run



ICP4D/WSL – Lifecycle Overview

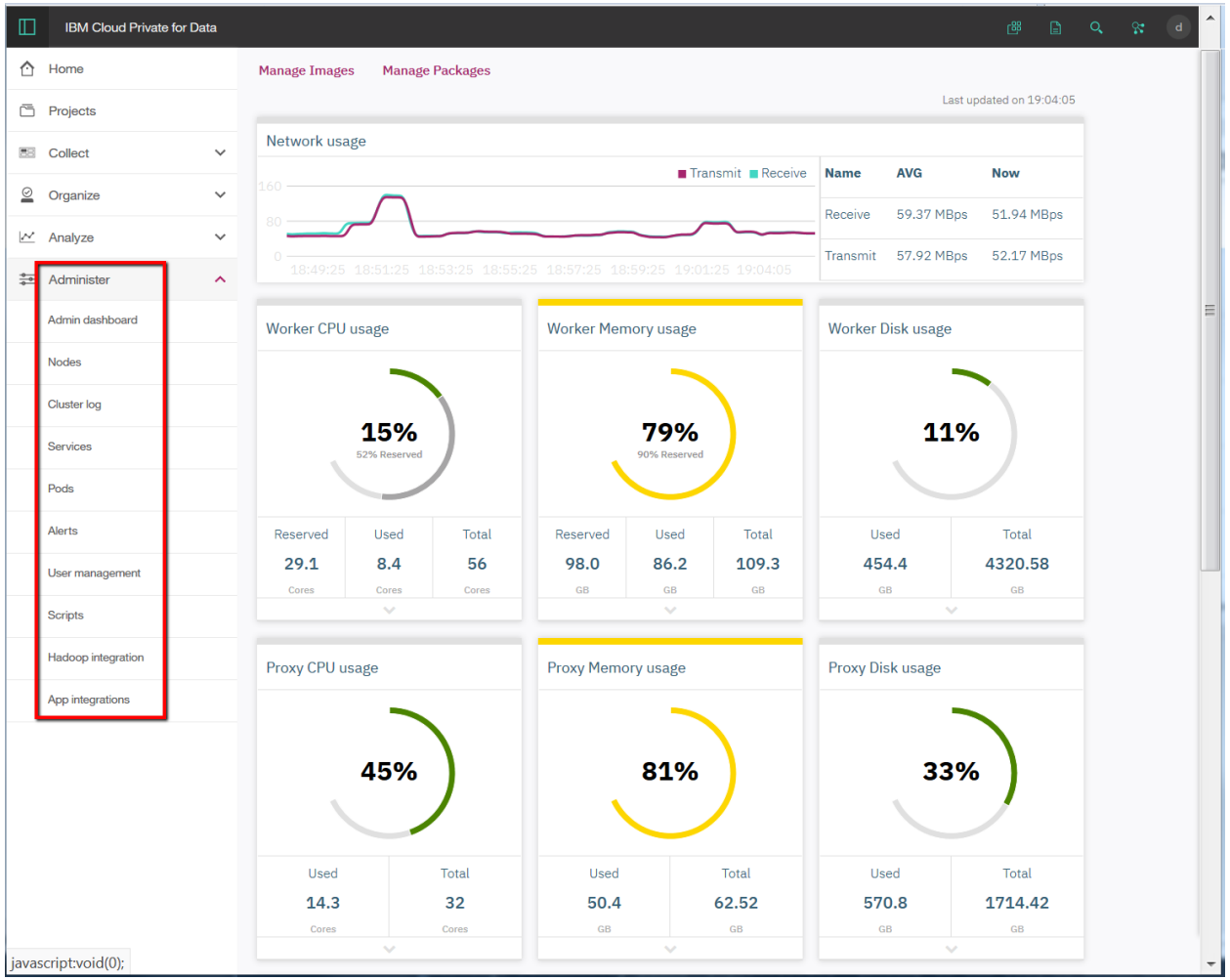
6b. Run Deployed Job Example – API Tab

- In the API tab, you can enter the function name and testing input, then click on submit



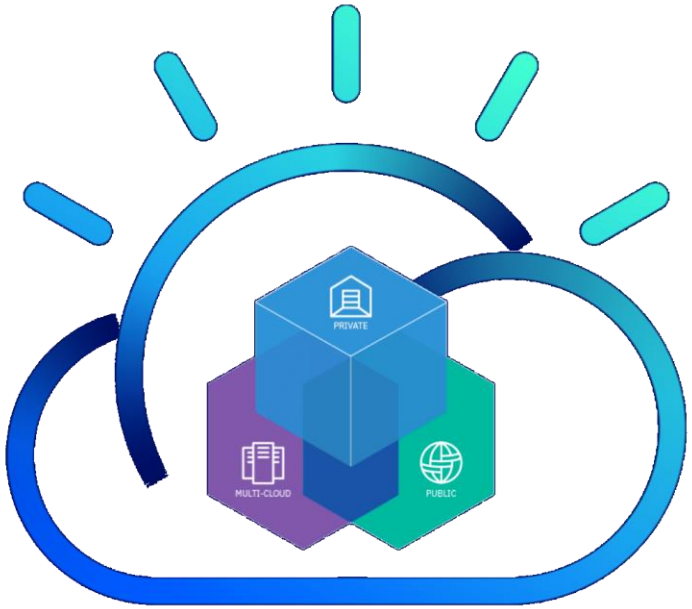
ICP4D/WSL – Lifecycle Overview

7. Admin Dashboard



IBM Cloud Private for Data / Watson Studio Local

Hadoop Integration



Best Practices

Access large amount of data – with Spark and Hadoop

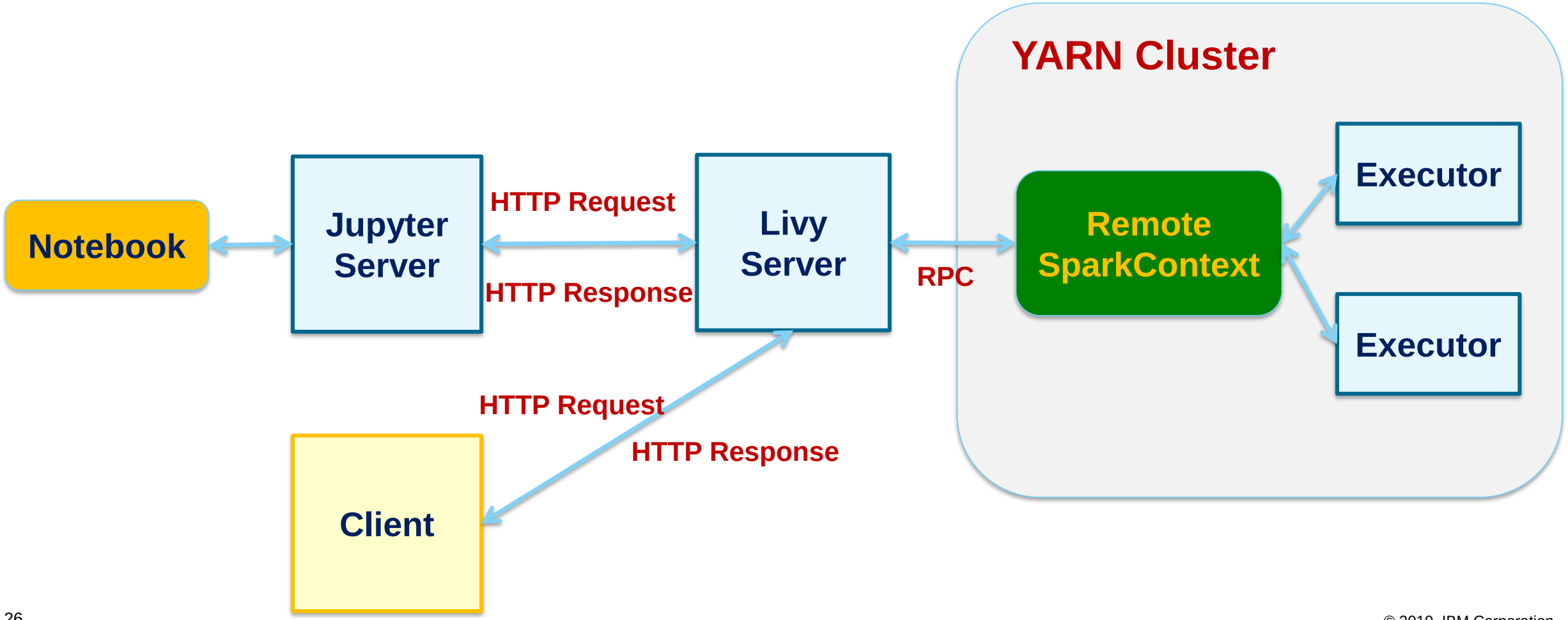
- As a best practice, processing should be close to your data. If you need to use large amount of data in HDFS or Hive, It's best to use the Spark running on Hadoop cluster through Livy.
- Take advantage of data localization, and avoids data transfer to speed up your processing.
- All runtime environments in WSL & ICP4D (Jupyter, Zeppelin and RStudio) support accessing Hadoop Spark through Livy.

Objective Overview

- Hadoop infrastructure is a huge investment but the tools in the ecosystem are not intuitive for business analyst and users. Most of your data is probably available within your HDFS along with your huge compute capacity but access to it is difficult.
- Build machine learning and analytics on Watson Studio that provides variety of open source libraries and frameworks such as Spark, R, Keras, TensorFlow, Scikit-learn as a service within a kubernetes based platform behind your firewall. Along with cloud agility the platform provides end-end data management and analytics capability with collaboration and lifecycle management functions.
- Hadoop connector provides seamless access to compute power available along with easy access to hdfs data.
- Accelerate your machine learning deployments in hours as oppose to months. Build quick to market business decision making Artificial Intelligence systems easily and quickly.

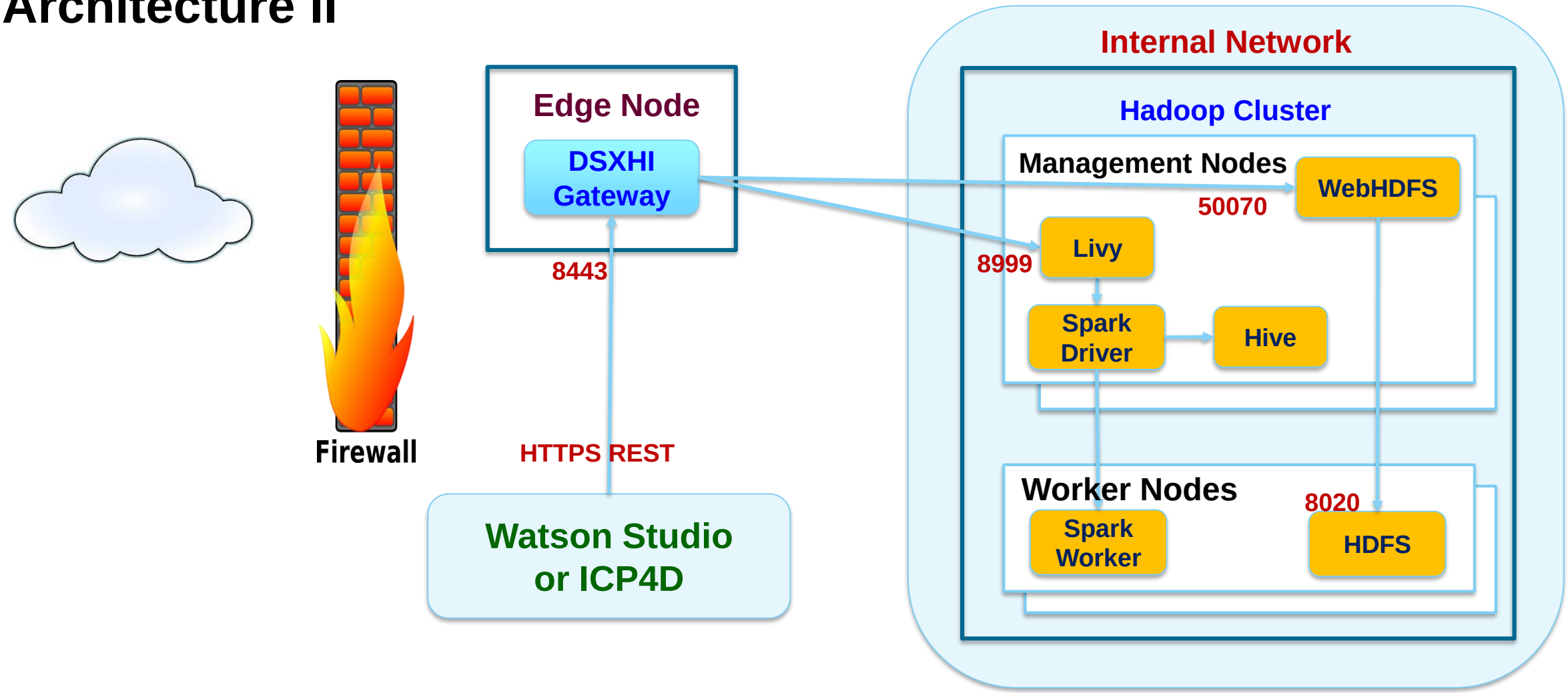
Livy Interface with Spark

Architecture I



Hadoop Connector Installation & Setup

Architecture II



Hadoop Gateway Installation & Setup

Installation

- **Install Livy connector (dsxhi) rpm package**
  - Download connector rpm file from Passport Advantage
  - Install rpm file (Ex. rpm -ivh dsxhi-icp4data-dsp-1.1.1.0-64.noarch.rpm)
- **Configure Livy connector**
  - Modify configuration files
    - ✓ # cp /opt/ibm/dsxhi/conf/dsxhi_install.conf.template.HDP /opt/ibm/dsxhi/conf/dsxhi_install.conf
    - ✓ # vi /opt/ibm/dsxhi/conf/dsxhi_install.conf

Hadoop Gateway Installation & Setup

Installation

- **Install Livy connector (dsxhi)**
  - Install connector

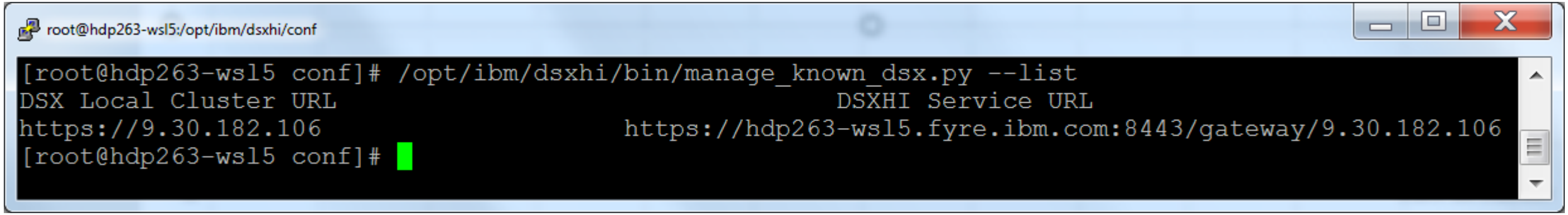
    # /opt/ibm/dsxhi/bin/install.py --dsxhi_gateway_master_password=<password> --password=<password>

```
IBM Data Science Experience Hadoop Integration Service
------------------------------------------------------
Terms and Conditions: http://www14.software.ibm.com/cgi-
bin/weblap/lap.pl?la_formnum=&li_formnum=L-KLSY-
AVZW2D&title=IBM+Data+Science+Experience+Hadoop+Integration&l=en

--Determining properties
--Running the prechecks

--Install Livy for Spark / Spark 2
--Configure gateway
--Create template for gateway
--Setting up known DSX Local clusters
--Install dsxhi_rest
--Start all services
--Install finished. Check status in /var/log/dsxhi/dsxhi.log
```

Hadoop Gateway Installation & Setup

Verify Installation

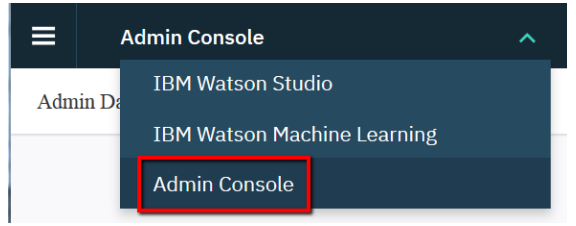
- **Livy connector process status check**
  - /opt/ibm/dsxhi/bin/status.py
- **List registered WSL cluster(s)**
  - /opt/ibm/dsxhi/bin/manage_known_dsx.py –list
- **Register/Add new WSL cluster**
  - /opt/ibm/dsxhi/bin/manage_known_dsx.py --add https://<WSL_FQDN>

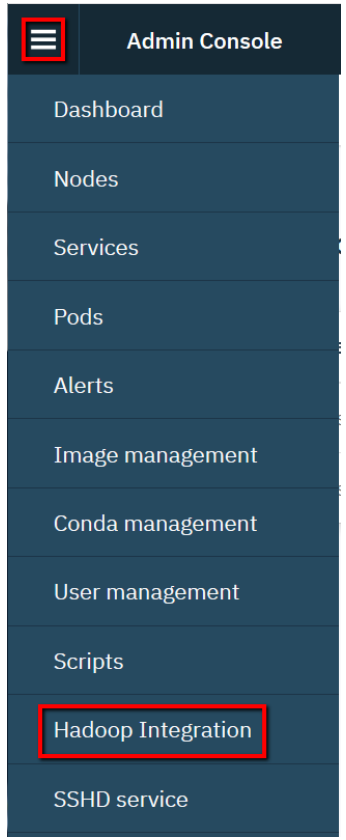


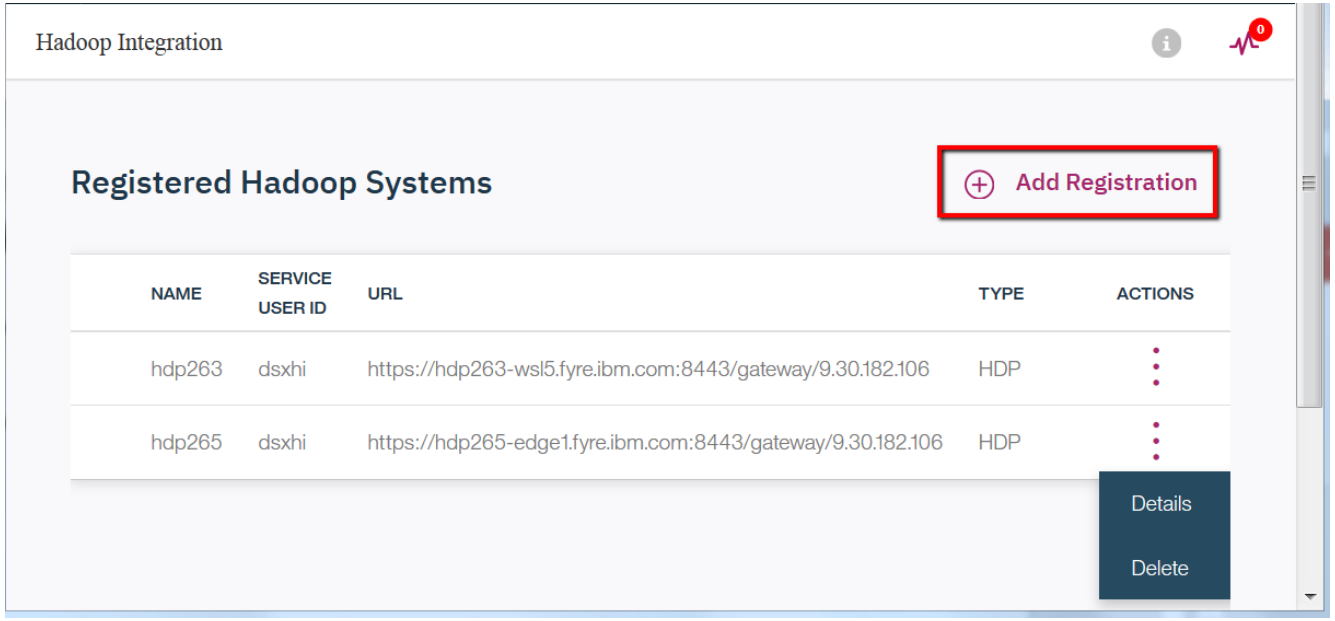
Hadoop Gateway Installation & Setup

Register Hadoop System in Admin Console

- Select “Admin Console” from Top drop down
- In the Admin Console, click the menu icon ( ) and click Hadoop Integration to view current registered Hadoop Cluster and/or register new Hadoop Clusters







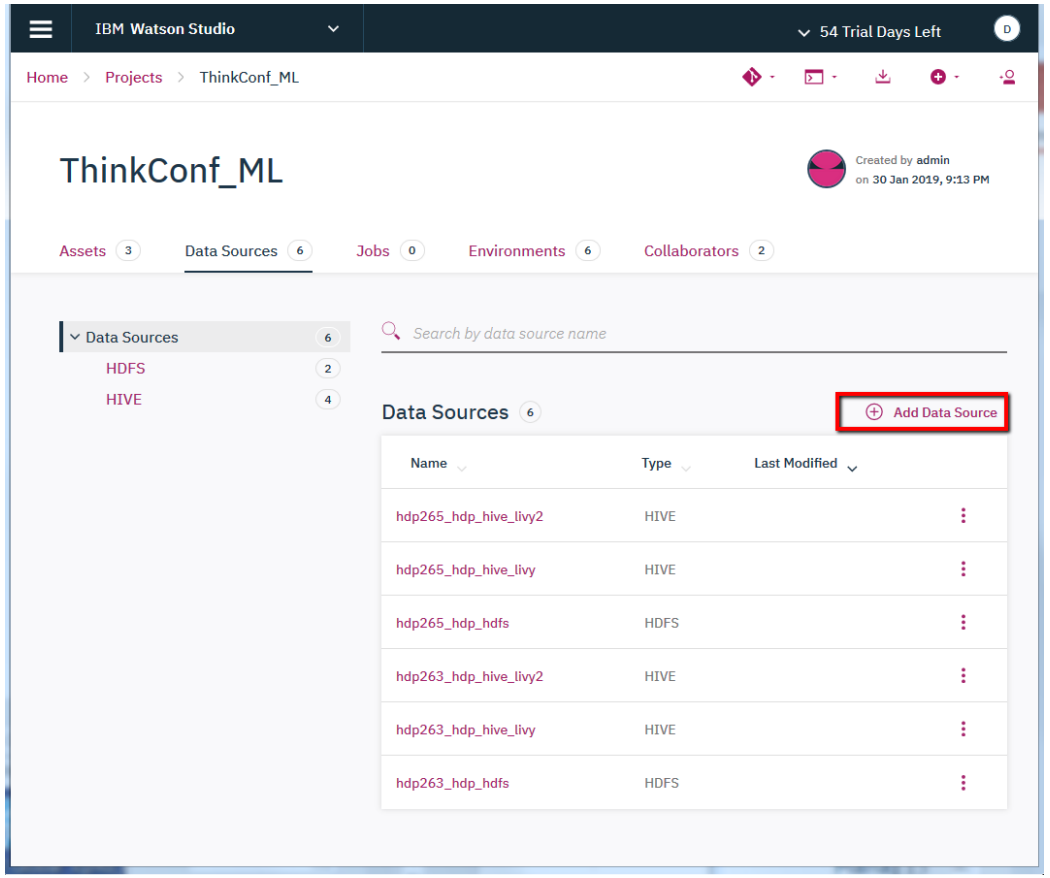
Hadoop Gateway Testing and Troubleshoot

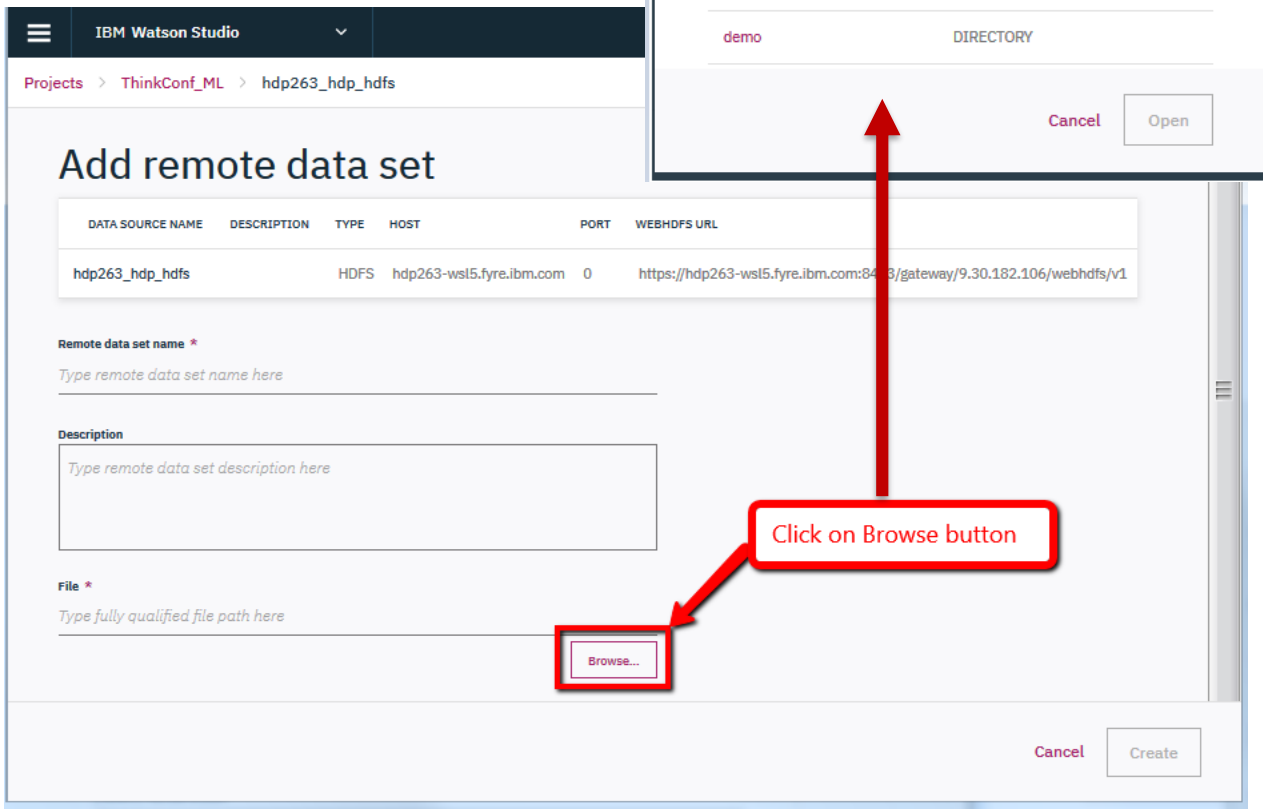
- Retrieve data from HDFS file system
- Access to Hive database table data
- Push down Spark job/task to the remote Spark2 server on Hadoop cluster

Hadoop Gateway – Example Patterns

1. Retrieving data from HDFS file system

- Add a remote data source from project dashboard

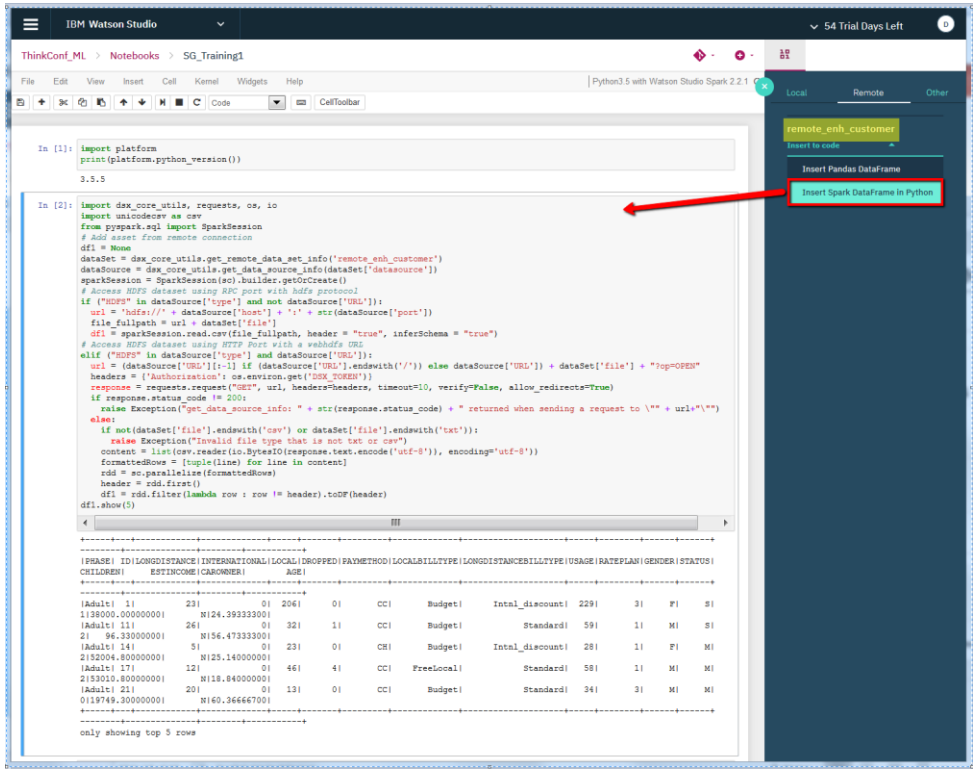




Hadoop Gateway – Example Patterns

2. Use added remote HDFS data source in Notebooks

- Click on icon ( ) and select the Remote data source created from previous instruction; then select “Insert Spark DataFrame in Python”.



Hadoop Gateway – Example Patterns

3. Load HDFS dataset via remote Spark session

- Example code:

```
import dsx_core_utils, requests, jaydebeapi, os, io, sys
import pandas as pd
url = 'https://hdp263-
wsl5.fyre.ibm.com:8443/gateway/9.30.182.106/webhdfs/v1/data/demo/customer/enhanced_customer.csv?OP=OPEN'
headers = {'Authorization': os.environ.get('DSX_TOKEN')}
response = requests.request("GET", url, headers=headers, timeout=10, verify=False, allow_redirects=True)
df = pd.read_csv(io.StringIO(response.text, newline=None), sep=',')
df.head()
```

In [6]:

```
import dsx_core_utils, requests, jaydebeapi, os, io, sys
import pandas as pd
url = 'https://hdp263-wsl5.fyre.ibm.com:8443/gateway/9.30.182.106/webhdfs/v1/data/demo/customer/enhanced_customer.csv?OP=OPEN'
headers = {'Authorization': os.environ.get('DSX_TOKEN')}
response = requests.request("GET", url, headers=headers, timeout=10, verify=False, allow_redirects=True)
df = pd.read_csv(io.StringIO(response.text, newline=None), sep=',')
df.head()
```

Out[6]:

	PHASE	ID	LONGDISTANCE	INTERNATIONAL	LOCAL	DROPPED	PAYMETHOD	LOCALBILLTYPE	LONGDISTANCEBILLTYPE	USAGE	RATEPLAN	GENDER
0	Adult	1	23	0	206	0	CC	Budget	Intnl_discount	229	3	F
1	Adult	11	26	0	32	1	CC	Budget	Standard	59	1	M
2	Adult	14	5	0	23	0	CH	Budget	Intnl_discount	28	1	F
3	Adult	17	12	0	46	4	CC	FreeLocal	Standard	58	1	M
4	Adult	21	20	0	13	0	CC	Budget	Standard	34	3	M

Hadoop Gateway – Example Patterns

4. Load HDFS dataset via “PyWebHdfs” package

```
import json
from pywebhdfs.webhdfs import PyWebHdfsClient
headers = {'Authorization': os.environ.get('DSX_TOKEN')}
url = 'https://hdp263-wsl5.fyre.ibm.com:8443/gateway/9.30.182.106/webhdfs/v1'
hdfs = PyWebHdfsClient(base_uri_pattern=url, request_extra_opts={'verify': False}, request_extra_headers=head
print(json.dumps(hdfs.list_dir('/'), indent=1))
```

```
In [19]: import json
         from pywebhdfs.webhdfs import PyWebHdfsClient
         headers = {'Authorization': os.environ.get('DSX_TOKEN')}

         url = 'https://hdp263-wsl5.fyre.ibm.com:8443/gateway/9.30.182.106/webhdfs/v1'
         hdfs = PyWebHdfsClient(base_uri_pattern=url, request_extra_opts={'verify': False}, request_extra_headers=headers)
         print(json.dumps(hdfs.list_dir('/'), indent=1))
```


```
{
 "FileStatuses": {
  "FileStatus": [
   {
    "group": "hadoop",
    "modificationTime": 1548968993029,
    "permission": "777",
    "pathSuffix": "app-logs",
    "type": "DIRECTORY",
    "fileId": 16396,
    "blockSize": 0,
    "accessTime": 0,
    "storagePolicy": 0,
    "owner": "yarn",
    "length": 0,
    "replication": 0,
    "childrenNum": 4
   },
   {
```

Hadoop Gateway – Example Patterns

5. Access Hive data

- %%spark -s <session_name> -c sql -o <local_dataframe>
  select gender, count(gender) from ext_customer group by gender order by gender
  print(type(df_ex_cust_local))
  print(df_ex_cust_local)

In [9]:
```
%%spark -s $session_name -c sql
show tables
```

Out[9]:

	database	tableName	isTemporary
0	default	del2	False
1	default	ext_customer	False

In [12]:
```
%%spark -s $session_name -o df_ex_cust_local -c sql
select gender, count(gender) from ext_customer group by gender order by gender
```

Out[12]:

	gender	count(gender)
0	F	1316
1	GENDER	1
2	M	750

In [15]:
```
print(type(df_ex_cust_local))
print(df_ex_cust_local)
```

```
<class 'pandas.core.frame.DataFrame'>
   gender  count(gender)
0       F           1316
1  GENDER              1
2       M            750
```

Hadoop Gateway – Remote Spark Execution

6. Push down Spark processing

In [25]:
```
import dsx_core_utils
%load_ext sparkmagic.magics
# Retrieve a list of registered Hadoop Integration systems.
DSXHI_SYSTEMS = dsx_core_utils.get_dsxhi_info(showSummary=True)
```

```
The sparkmagic.magics extension is already loaded. To reload it, use:
  %reload_ext sparkmagic.magics
Available Hadoop systems:

  systemName  LIVYSPARK  LIVYSPARK2 imageId
0     hdp263  livyspark  livyspark2
1     hdp265  livyspark  livyspark2
```

In [26]:
```
# Set up sparkmagic to connect to the selected registered HI systemName above.
myConfig={
"queue": "default",
"driverMemory": "512M",
"numExecutors": 1,
"executorMemory":"512M"
}
HI_CONFIG = dsx_core_utils.setup_livy_sparkmagic(
system="hdp263",
livy="livyspark2",
imageId=None,
addlConfig=myConfig)
# (Re-)load spark magic to apply the new configs.
%reload_ext sparkmagic.magics
```

```
sparkmagic has been configured to use https://hdp263-wsl5.fyre.ibm.com:8443/gateway/9.30.182.106/livy2/v1
success configuring sparkmagic livy.
```

Hadoop Gateway – Remote Spark Execution

6. Push down Spark processing

In [27]:
```
%spark cleanup
session_name = 'spark2_0205a'
livy_endpoint = HI_CONFIG['LIVY']
webhdfs_endpoint = HI_CONFIG['WEBHDFS']
%spark add -s $session_name -l python -u $livy_endpoint
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
4	application_1549042673081_0007	pyspark	idle	Link	Link	✓

```
SparkSession available as 'spark'.
```

In [28]:
```
%spark info
```

```
Info for running Spark:
    Sessions:
        Name: spark2_0205a      Session id: 4   YARN id: application_1549042673081_0007 Kind: pyspark   State: idle
        Spark UI: http://hdp263-wsl2.fyre.ibm.com:8088/proxy/application_1549042673081_0007/
        Driver Log: http://hdp263-wsl5.fyre.ibm.com:8042/node/containerlogs/container_e02_1549042673081_0007_01_000001/dsxhi
    Session configs:
        {'conf': {'spark.yarn.appMasterEnv.HI_UTILS_PATH': '/user/dsxhi/environments/pythonAddons/hi_core_utils.zip'}, 'proxyUser': 'ds
xhi', 'numExecutors': 1, 'queue': 'default', 'driverMemory': '512M', 'executorMemory': '512M'}
```

Hadoop Gateway – Remote Spark Execution

6. Push down Spark processing

In [30]:
```
%%spark -s $session_name
import socket
print("Remote livy session driver: {}".format(socket.gethostname()))
```

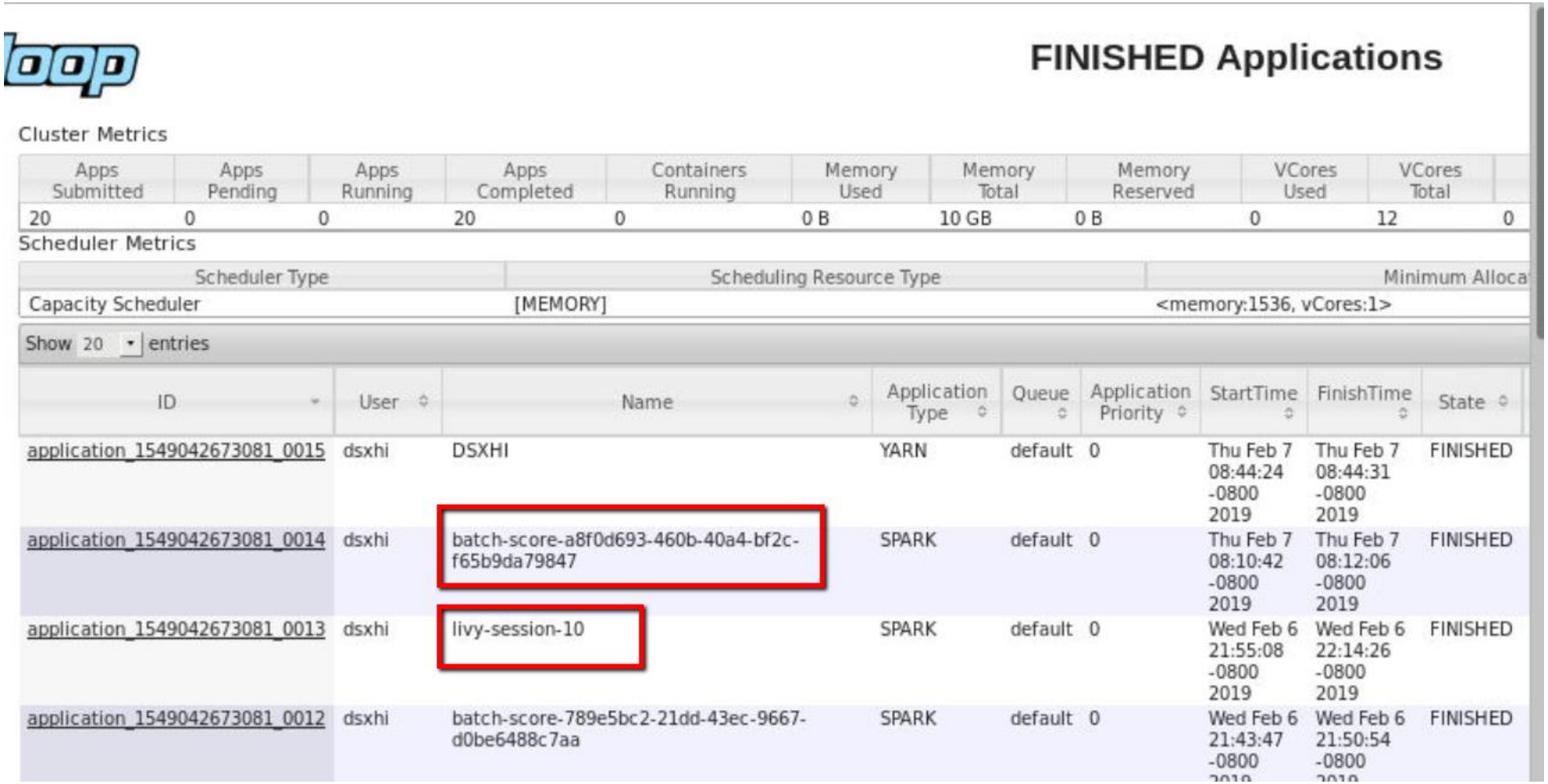
```
Remote livy session driver: hdp263-wsl5.fyre.ibm.com
```

In [32]:
```
%%spark

file_customer = "hdfs:///data/demo/customer/enhanced_customer.csv"
df_cust = spark.read.format("org.apache.spark.sql.execution.datasources.csv.CSVFileFormat").option("header", "true").option("inferSchema", "true").load(file_customer)
df_cust.printSchema()

print("Total number of customer dataset row count: {}\n".format(df_cust.count()))
```

```
root
 |-- PHASE: string (nullable = true)
 |-- ID: integer (nullable = true)
 |-- LONGDISTANCE: integer (nullable = true)
 |-- INTERNATIONAL: integer (nullable = true)
 |-- LOCAL: integer (nullable = true)
 |-- DROPPED: integer (nullable = true)
 |-- PAYMETHOD: string (nullable = true)
 |-- LOCALBILLTYPE: string (nullable = true)
 |-- LONGDISTANCEBILLTYPE: string (nullable = true)
 |-- USAGE: integer (nullable = true)
 |-- RATEPLAN: integer (nullable = true)
 |-- GENDER: string (nullable = true)
 |-- STATUS: string (nullable = true)
 |-- CHILDREN: integer (nullable = true)
 |-- ESTINCOME: double (nullable = true)
 |-- CAROWNER: string (nullable = true)
 |-- AGE: double (nullable = true)

Total number of customer dataset row count: 2066
```

Hadoop Gateway – Remote Spark Execution

6. Push down Spark processing – Review the remote Spark application



loop

FINISHED Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	
20	0	0	20	0	0 B	10 GB	0 B	0	12	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Alloca
Capacity Scheduler	[MEMORY]	<memory:1536, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State
application_1549042673081_0015	dsxhi	DSXHI	YARN	default	0	Thu Feb 7 08:44:24 -0800 2019	Thu Feb 7 08:44:31 -0800 2019	FINISHED
application_1549042673081_0014	dsxhi	batch-score-a8f0d693-460b-40a4-bf2c-f65b9da79847	SPARK	default	0	Thu Feb 7 08:10:42 -0800 2019	Thu Feb 7 08:12:06 -0800 2019	FINISHED
application_1549042673081_0013	dsxhi	livy-session-10	SPARK	default	0	Wed Feb 6 21:55:08 -0800 2019	Wed Feb 6 22:14:26 -0800 2019	FINISHED
application_1549042673081_0012	dsxhi	batch-score-789e5bc2-21dd-43ec-9667-d0be6488c7aa	SPARK	default	0	Wed Feb 6 21:43:47 -0800	Wed Feb 6 21:50:54 -0800	FINISHED

Hadoop Gateway – Remote Spark Execution

Delete the remote Spark session**

- %spark delete -s <session_name>

```
In [38]: %spark delete -s $session_name
```

```
In [23]: %spark cleanup
```

```
In [*]: %%javascript
Jupyter.notebook.session.delete();
```

For additional learning

- Kubernetes - https://kubernetes.io/docs/concepts/
- Dockers and containers - https://docs.docker.com/engine/docker-overview/
- IBM Cloud Private documentation : https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0.3/getting_started/introduction.html
- IBM Cloud Private for Data documentation : https://docs-icpdata.mybluemix.net/docs/content/com.ibm.icpdata.doc/zen/overview/overview.html

Backup Slides