Using R in Scalable Compute Contexts – R Server with Spark on HDInsight

Ali Zaidi
Microsoft Machine Learning and Data Science Team
learnanalytics.microsoft.com

Microsoft

R – What is it?

Open Source
"lingua franca"

Analytics, Computing, Modeling

Global Community

Millions of users

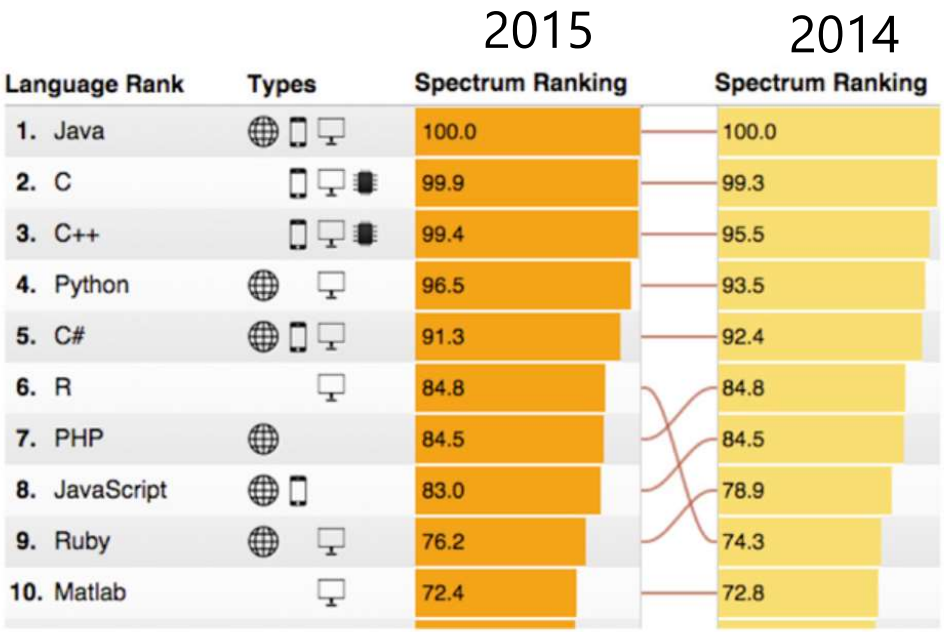
Ecosystem

7000+ Algorithms, Test Data & Evaluations

Common R use cases

Vertical	Sales & Marketing	Finance & Risk	Customer & Channel	Operations & Workforce
Retail	Demand Forecasting Loyalty Programs Cross-sell & Upsell Customer Acquisition	Fraud Detection Pricing Strategy	Personalization Lifetime Customer Value Product Segmentation	Store Location Demographics Supply Chain Management Inventory Management
Financial Services	Customer Churn Loyalty Programs Cross-sell & Upsell Customer Acquisition	Fraud Detection Risk& Compliance Loan Defaults	Personalization Lifetime Customer Value	Call Center Optimization Pay for Performance
Healthcare	Marketing Mix Optimization Patient Acquisition	Fraud Detection Bill Collection	Population Health Patient Demographics	Operational Efficiency Pay for Performance
Manufacturing	Demand Forecasting Marketing mix Optimization	Pricing Strategy Perf Risk Management	Supply Chain Optimization Personalization	Remote Monitoring Predictive Maintenance Asset Management

R has some challenges

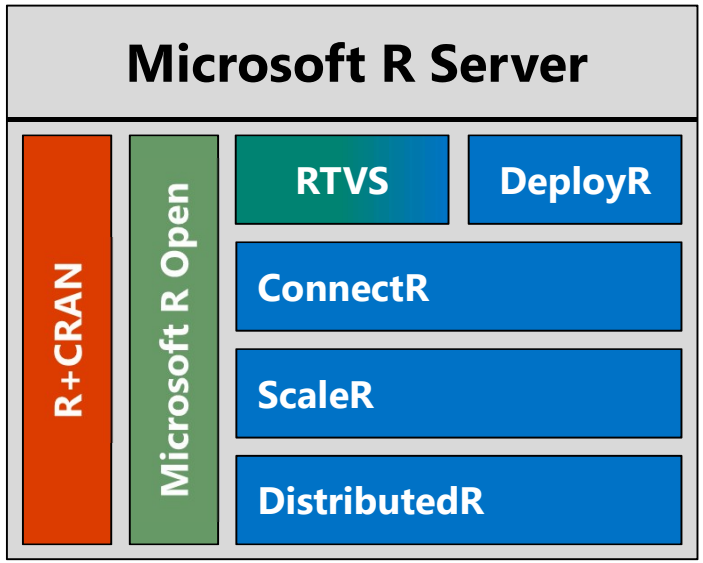


Language Rank	Types	2015 Spectrum Ranking	2014 Spectrum Ranking
1. Java		100.0	100.0
2. C		99.9	99.3
3. C++		99.4	95.5
4. Python		96.5	93.5
5. C#		91.3	92.4
6. R		84.8	84.8
7. PHP		84.5	84.5
8. JavaScript		83.0	78.9
9. Ruby		76.2	74.3
10. Matlab		72.4	72.8

IEEE Spectrum July 2015

- **Data Flows Overwhelm Open Source R**
  - In-Memory Operation
  - Lack of Implicit Parallelism
  - Expensive Data Movement & Duplication
- **Not enterprise ready**
  - Inadequacy of Community Support
  - Lack of Guaranteed Support Timeliness
  - No SLAs or Support models

R Server: scale-out R, Enterprise Class!



- Enterprise Scale & Performance
  - Scales from workstations to large clusters
  - Can process terabytes of data faster
  - Growing portfolio of Parallelized algorithms
- Enterprise Class Support
- Secure R Deployment/Operationalization
- Write Once Deploy Anywhere for multiple platforms
  - Cloud: HDInsight and Marketplace
  - On Prem: SQL Server, Hadoop (HortonWorks, Cloudera, MapR) and TeraData DB
- IDE for data scientists and developers (R Tools for Visual Studio)

R Server: scale-out R, Enterprise Class!

- 100% compatible with open source R
  - Any code/package that works today with R will work in R Server
- Wide range of scalable and distributed R functions
  - Examples: rxDataStep(), rxSummary(), rxGlm(), rxDForest(), rxPredict()
- Ability to parallelize any R function
  - Ideal for parameter sweeps, simulation or multiple runs

Parallelized & Distributed Algorithms

Data Step

- Data import – Delimited, Fixed, SAS, SPSS, OBDC
- Variable creation & transformation
- Recode variables
- Factor variables
- Missing value handling
- Sort, Merge, Split
- Aggregate by category (means, sums)

Descriptive Statistics

- Min / Max, Mean, Median (approx.)
- Quantiles (approx.)
- Standard Deviation
- Variance
- Correlation
- Covariance
- Sum of Squares (cross product matrix for set variables)
- Pairwise Cross tabs
- Risk Ratio & Odds Ratio
- Cross-Tabulation of Data (standard tables & long form)
- Marginal Summaries of Cross Tabulations

Statistical Tests

- Chi Square Test
- Kendall Rank Correlation
- Fisher's Exact Test
- Student's t-Test

Sampling

- Subsample (observations & variables)
- Random Sampling

Predictive Statistics

- Sum of Squares (cross product matrix for set variables)
- Multiple Linear Regression
- Generalized Linear Models (GLM) exponential family distributions: binomial, Gaussian, inverse Gaussian, Poisson, Tweedie. Standard link functions: cauchit, identity, log, logit, probit. User defined distributions & link functions.
- Covariance & Correlation Matrices
- Logistic Regression
- Predictions/scoring for models
- Residuals for all models

Variable Selection

- Stepwise Regression

Simulation

- Simulation (e.g. Monte Carlo)
- Parallel Random Number Generation

Cluster Analysis

- K-Means

Machine Learning

- Decision Trees
- Decision Forests
- Gradient Boosted Decision Trees
- Naïve Bayes

Custom Parallelization

- rxDataStep
- rxExec
- PEMA-R API

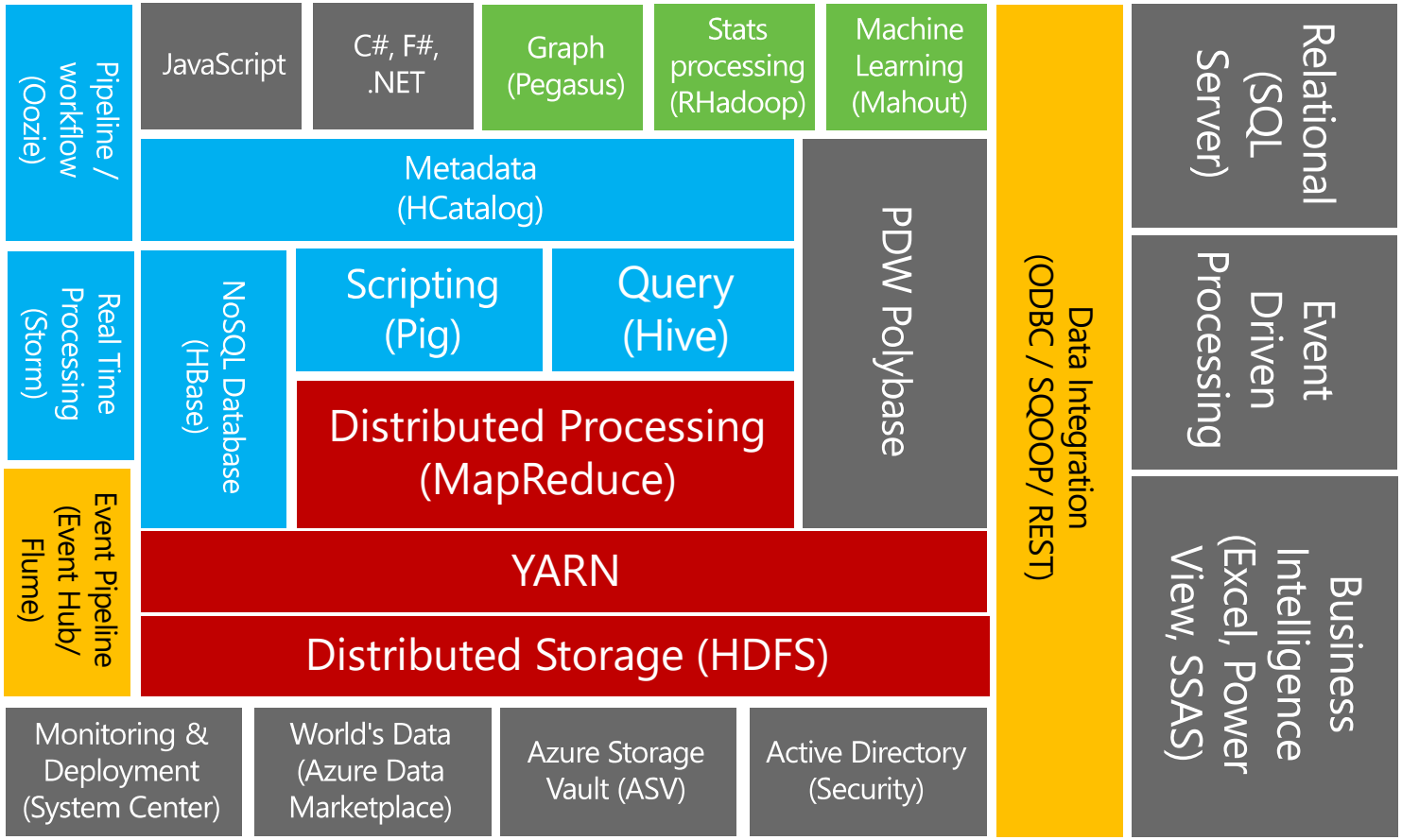
Hadoop and HDInsight

Using the Hadoop Ecosystem to process and query data

Hadoop

- An ecosystem of components for distributed data processing and analysis
- Core components: MapReduce, HDFS, YARN
- Data is processed in the Hadoop Distributed File System (HDFS)
- Resource Management is performed by YARN
- Many other related projects

HDInsight and the Hadoop ecosystem



Core Hadoop

Data processing

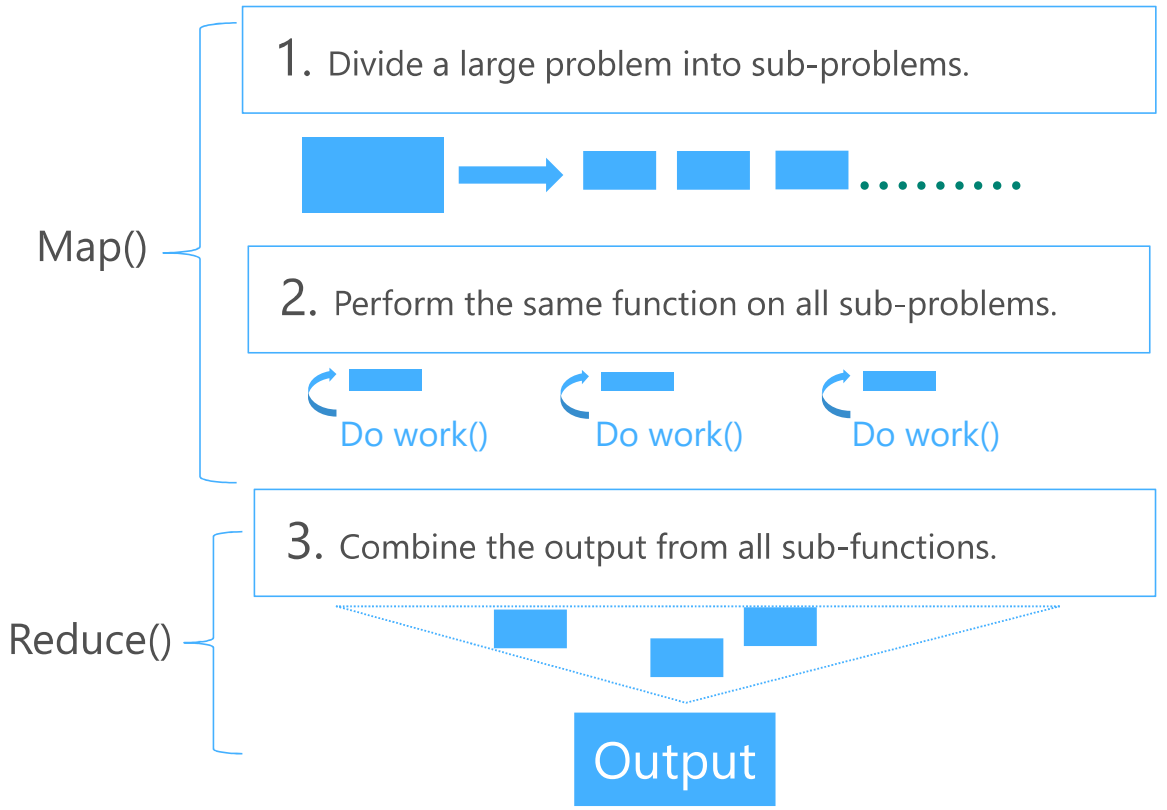
Microsoft integration

Data Movement

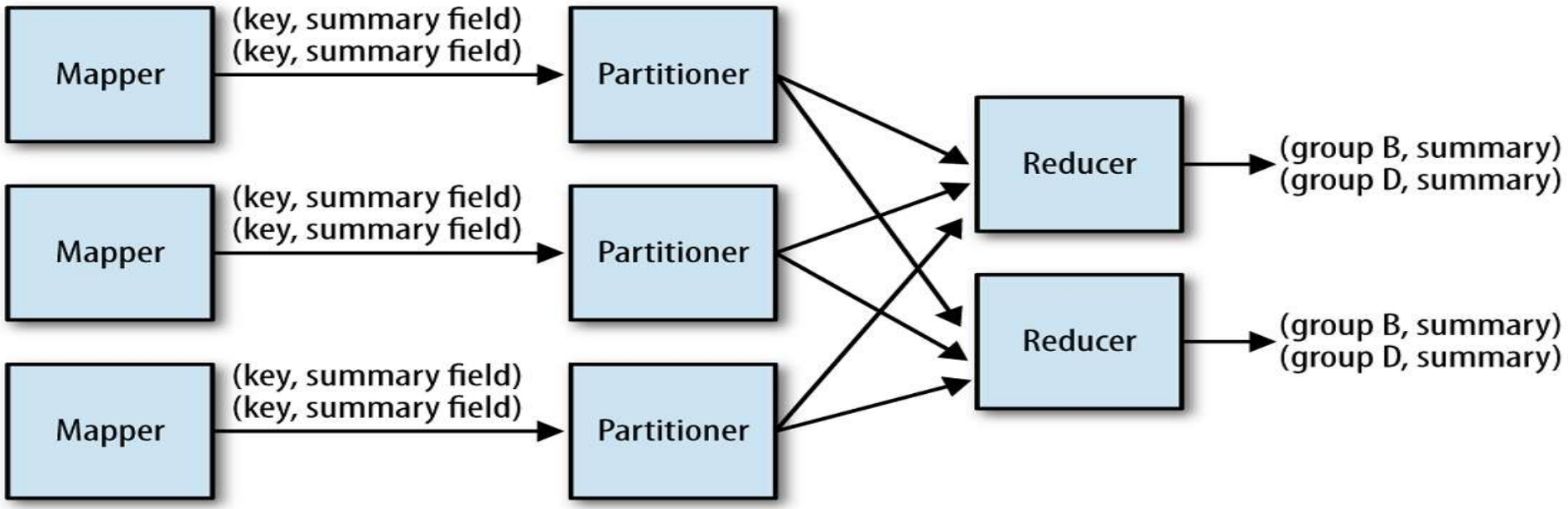
Packages

Hadoop MapReduce

- Programming framework (library and runtime) for analyzing datasets stored in HDFS
- Composed of user-supplied Map and Reduce functions:
  - Map() - subdivide and conquer
  - Reduce() - combine and reduce cardinality



Another view of MapReduce



HDInsight

- 3 Modes: VM, Service, On-Demand
- Azure Storage or Azure Data Lake provides the HDFS layer
- Azure SQL Database stores metadata

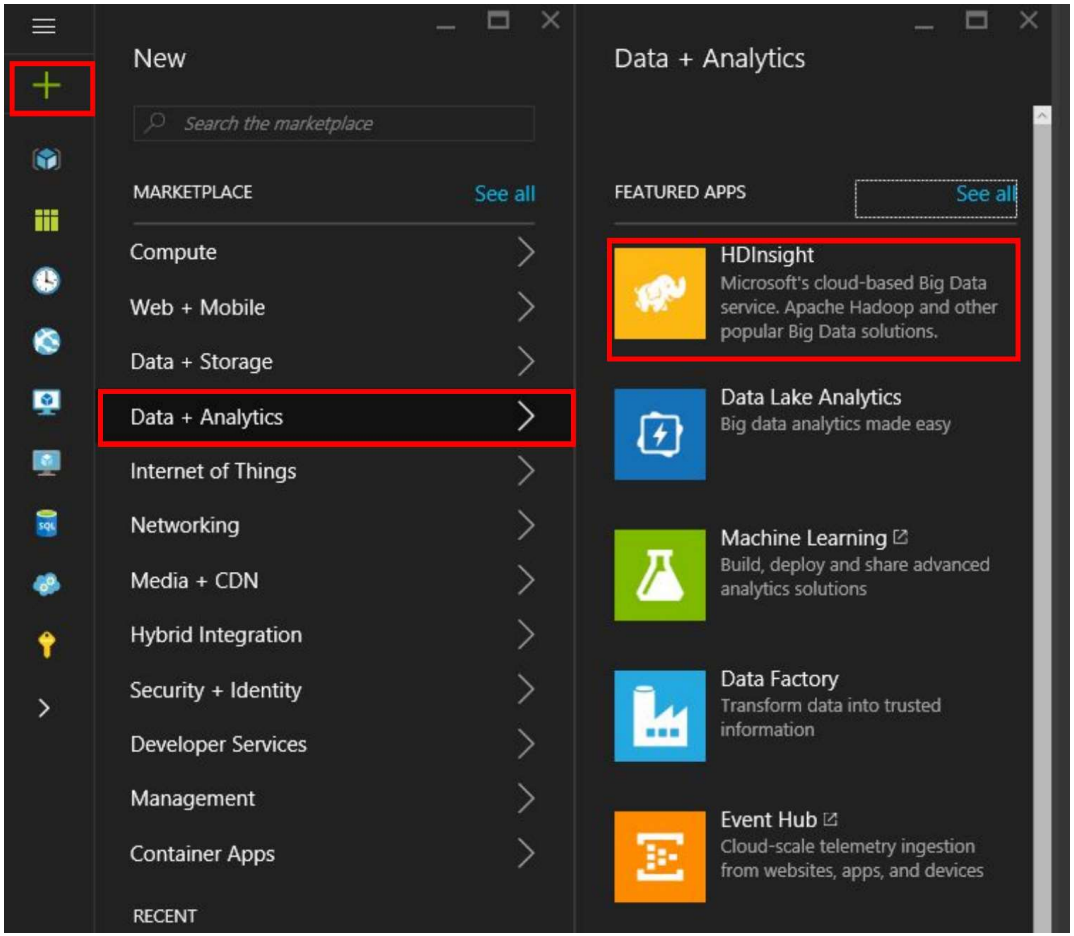
Deploying HDInsight Clusters

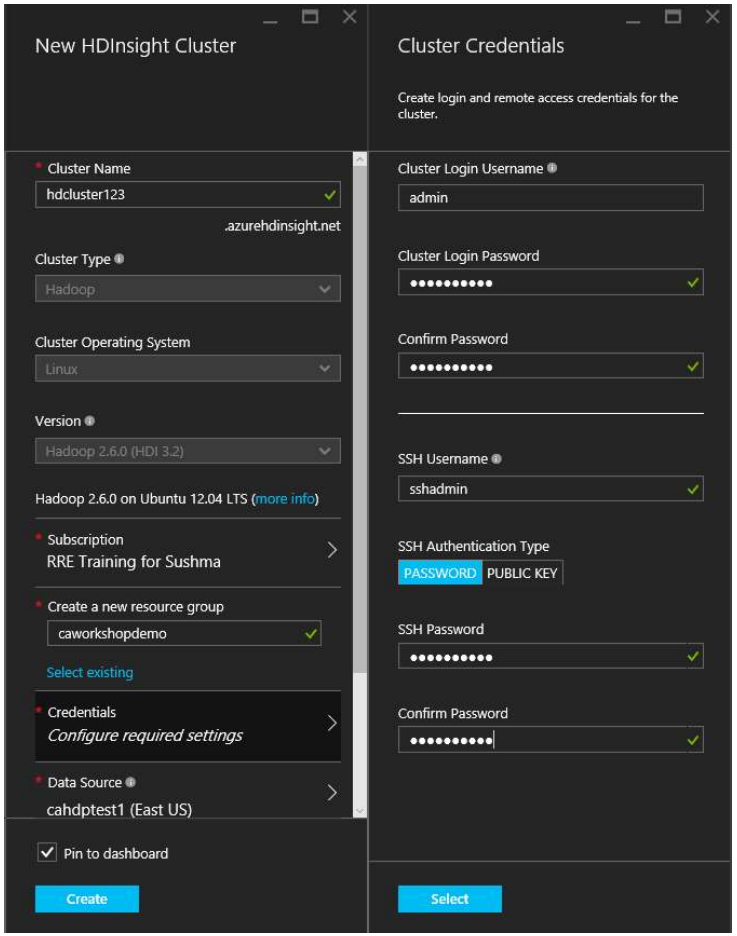
- Cluster Type: Hadoop, Spark, HBase and Storm.
  - Hadoop clusters: for query and analysis workloads
  - HBase clusters: for NoSQL workloads
  - Spark clusters: for in-memory processing, interactive queries, stream, and machine learning workloads
- Operating System: Windows or Linux
- Can be deployed from Azure portal, Azure Command Line Interface (CLI), or Azure PowerShell and Visual Studio
- A UI dashboard is provided to the cluster through Ambari.
- Remote Access through SSH, REST API, ODBC, JDBC.
  - Remote Desktop (RDP) access for Windows clusters

Components and Customization

- Script Actions can be run during cluster provisioning to install additional components.
- You can use the sample script actions during deployment to install:
  - Solr
  - R
  - Giraph
- You can change the configurations of a cluster using Bootstrap with:
  - Azure PowerShell
  - .NET SDK
  - ARM Templates

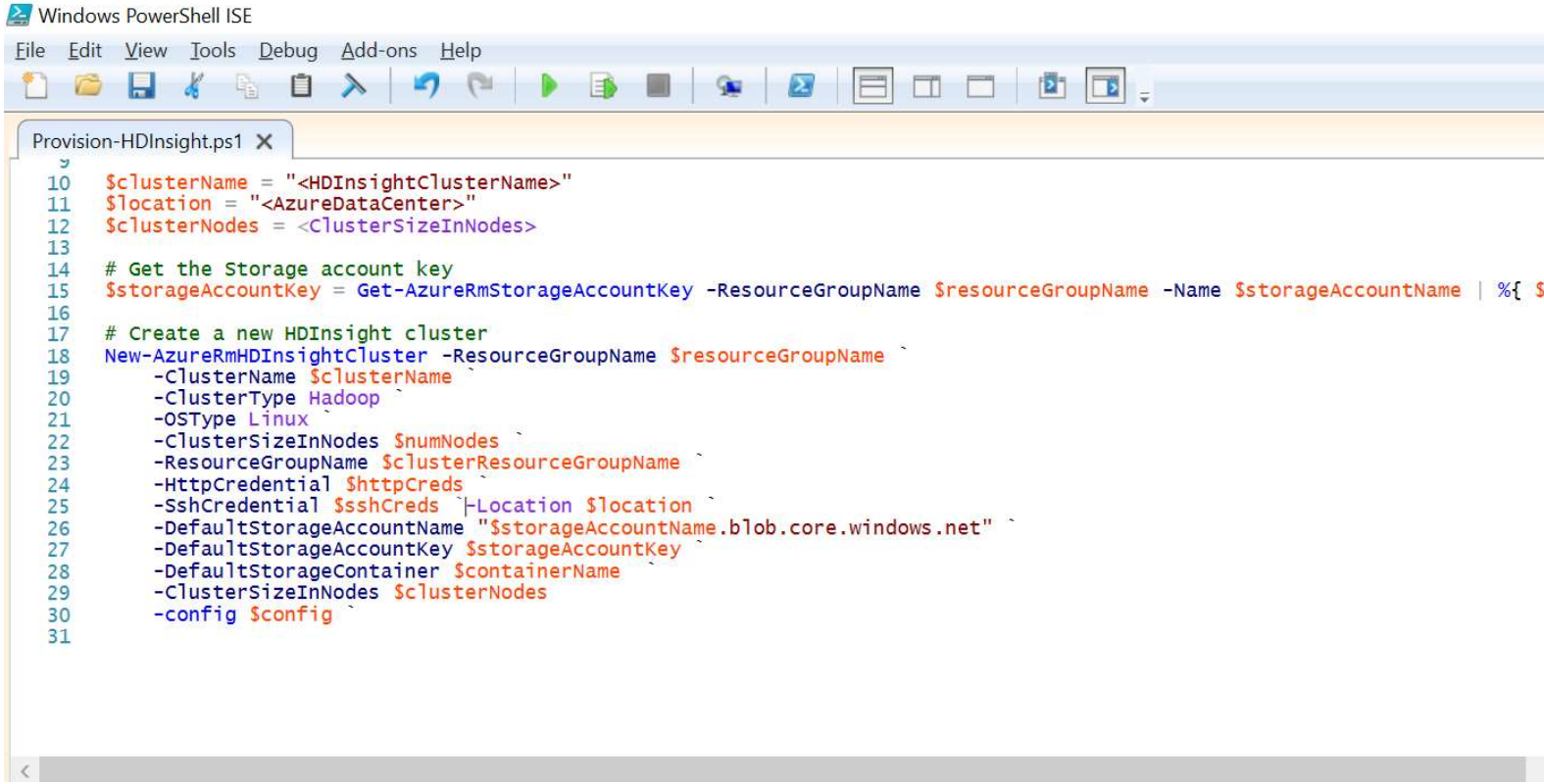
Provisioning with Azure Portal



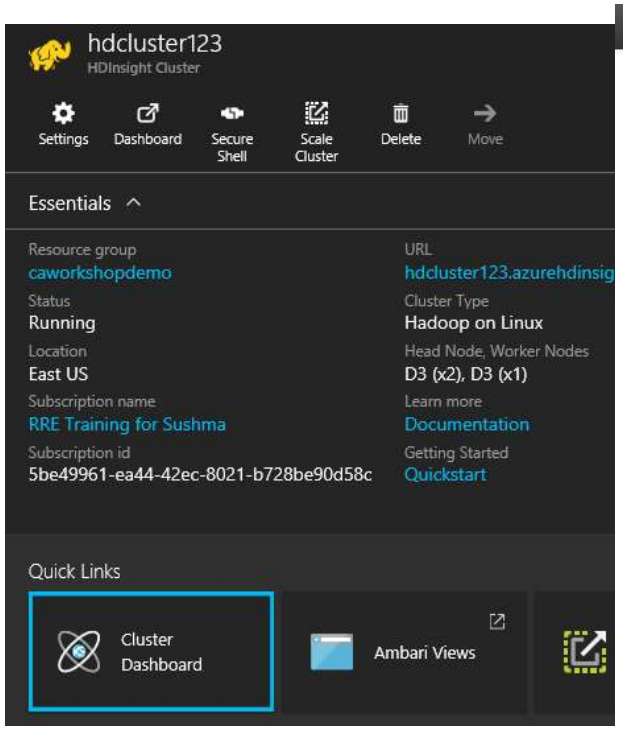


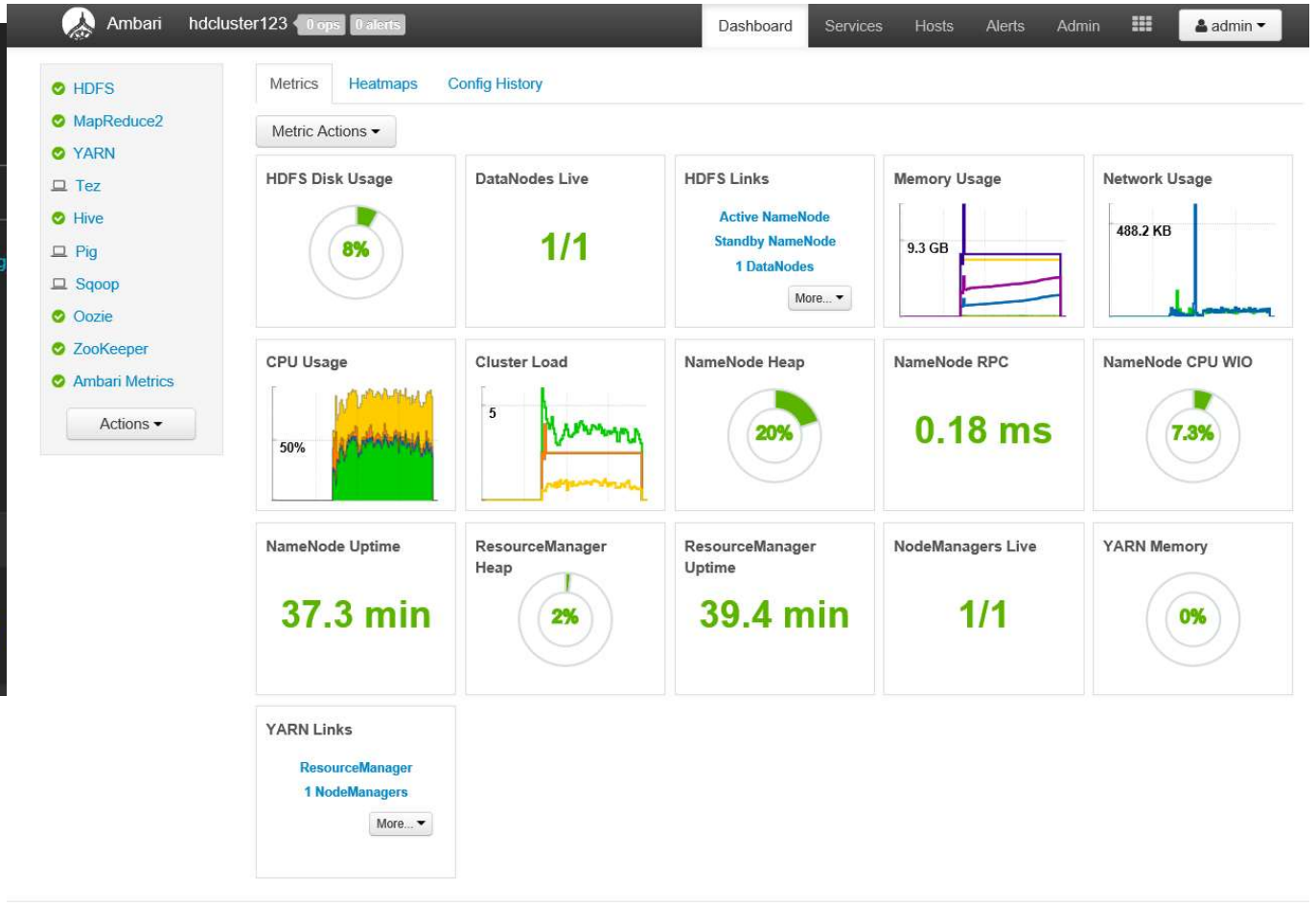
Provisioning Clusters Using PowerShell

- You can provision with Azure PowerShell

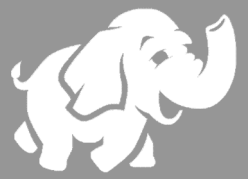


Cluster Dashboard Access - Ambari





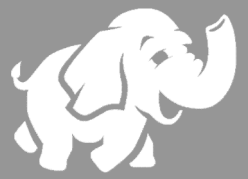
What Is Spark?



Spark is

- an open-source project that performs rapid calculations on in-memory datasets
- Started at the University of California Berkeley AMPLab
- Open Source [Apache hosted & licensed]
- Developed by a large community of developers (currently the most active Apache project)

R Server in Hadoop



- Traditionally R Server supported Hadoop MR as a Hadoop compute context
- This meant R Server functions ran on Hadoop as a MapReduce Job under YARN
- Spark posits itself as a better execution engine for iterative jobs
- R Server on MR is slowed by startup overhead and file-based communications

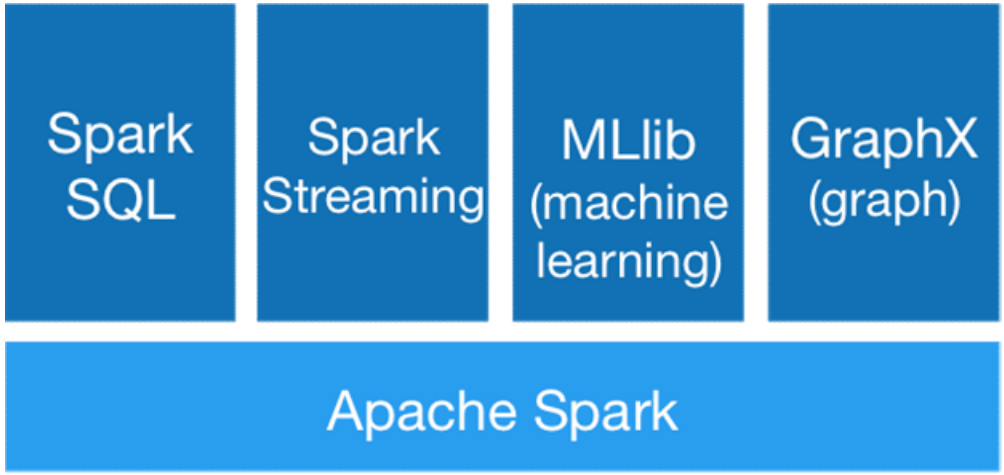
R Server on Spark

- Spark permits data to be retained in memory across iterations of ML pipelines through Resilient Distributed Datasets (RDD) and Spark Dataframes
- R Server on Spark is built on Spark's high performance distributed backend w/o the file-based overhead of MR
- How it works:
  - Use RDD's to load and retain data across iterations
  - Cache data in RDD between iterations
  - Master and worker nodes communicate with Scala API for communication
  - Saves temporary files in tmpfs (in-memory file system on Linux)
  - Supports persistent mode in Spark Context to reduce overhead of launching the process.

How to Write Spark Jobs – Spark Abstractions

- Spark supports popular language APIs such as: Scala, Python, R, Java
  - Spark SQL: Seamlessly mix SQL queries with Spark programs
  - Datasets – RDD
    - RDD (Resilient Distributed Data) is the basis for what Spark enables
      - Resilient – the models can be recreated on the fly from known state
      - Distributed – the dataset is often partitioned across multiple nodes for increased scalability and parallelism
- Data Scientists prefer to use DataFrames tabular derivative of RDDs

Spark Core



- Elegant Developer APIs: Data Frames/SQL, Machine Learning, Graph algorithms and streaming
- High Level API in Scala, Python, Java and R
- Promotes Code Reuse – APIs and data types are similar for batch and streaming jobs

- Workloads: MLLib and GraphX for Data Scientists, Spark SQL for Data Analysts, Spark Streaming for micro batch problems

Spark Supported Data Types

Built-In

{ JSON }

JDBC

Parquet

HIVE

MySQL

PostgreSQL

HDFS

amazon web services S3

H2

External

AVRO

CSV

dBase

APACHE HBASE

elasticsearch.

cassandra

amazon web services
Amazon Redshift

and more…

Spark v. MapReduce

- How??
  - Caching data to memory can avoid extra reads to disk
  - Scheduling of tasks from 15 – 20 s to 15 – 20 ms
  - Resources are dedicated to the entire life of the application
  - Can link multiple maps and reduces together without having to rewrite intermediate data to HDFS
  - Every reduce doesn't require a map

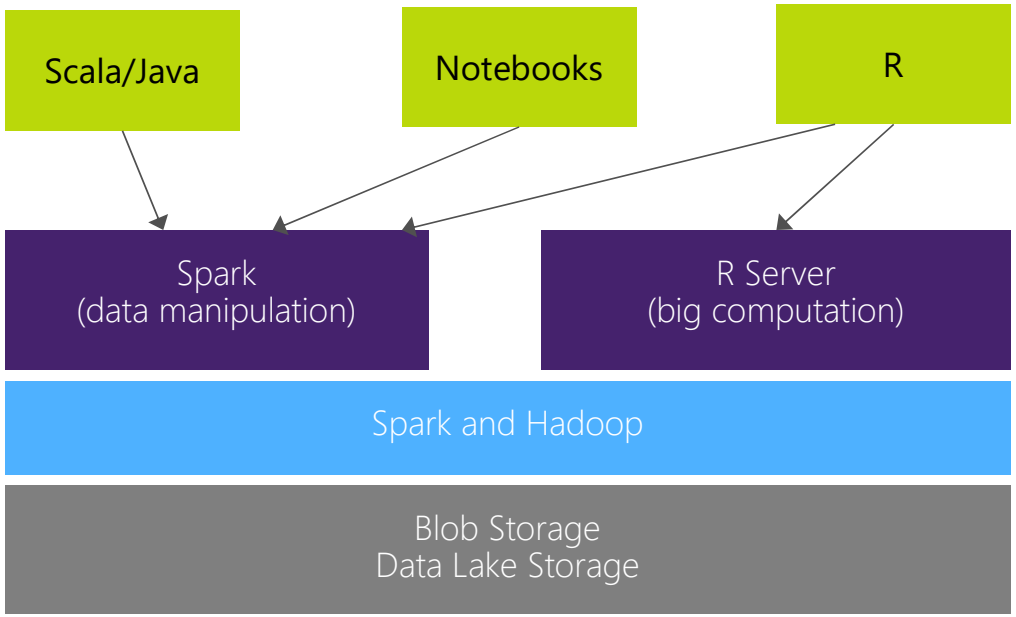
Spark Main Considerations

- Spark's Model of Parallel Computing
  - RDDs form the basis of data abstraction in Spark
    - RDDs are immutable distributed collections of objects
  - Spark, as a Scala derivative, is a functional programming paradigm
  - Functional abstractions are transformers and actions:

Applied to an RDD lazily, creating a promise to generate a new RDD

Triggers scheduler, retreives results

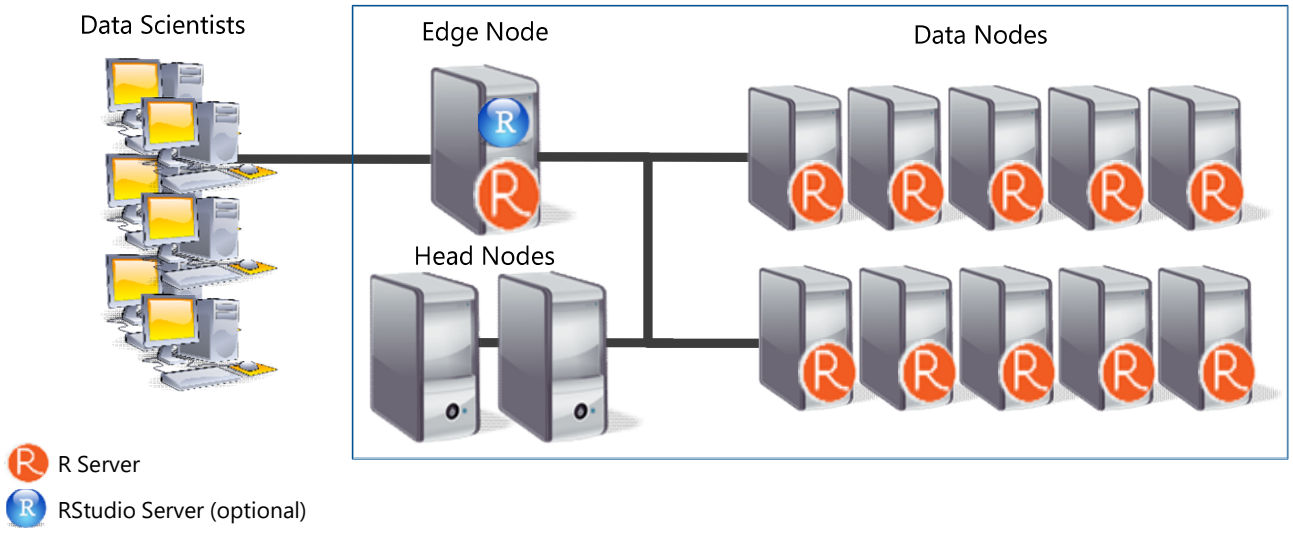
HDInsight + R Server: Managed Hadoop for advanced analytics



Hadoop: lingua franca for BigData

- Spark (Standard)
  - Integrated notebooks experience
  - Upgraded to latest Version 1.6
- R Server (Premium)
  - Leverage R skills with massively scalable algorithms and statistical functions
  - Reuse existing R functions over multiple machines

R Server HDInsight Architecture



R Server Distributed Processing:

Master R process on Edge node

↓

Apache YARN and Spark

↓

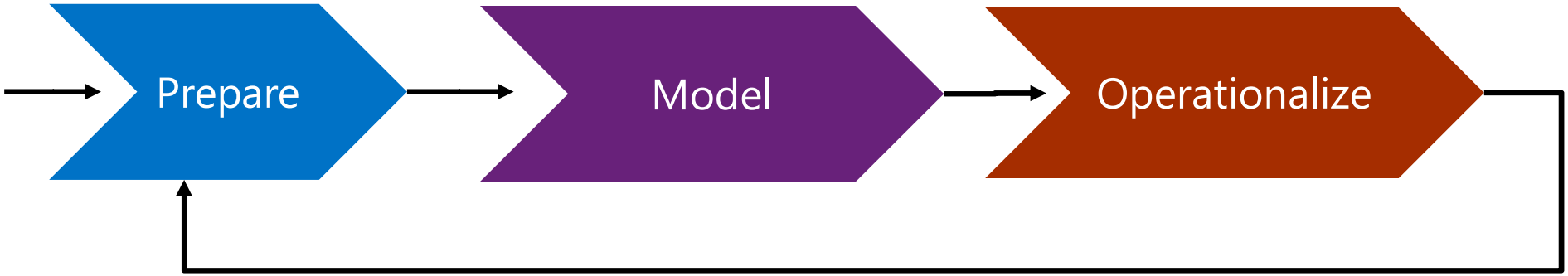
Worker R processes on Data nodes

Typical advanced analytics lifecycle

Prepare: Assemble, cleanse, profile and transform diverse data relevant to the subject.

Model: Use statistical and machine learning algorithms to build classifiers and make predictions

Operationalize: Apply predictions and visualizations to support business applications



Airline Arrival Delay Prediction Demo

- Clean/Join – Using SparkR from R Server
- Train/Score/Evaluate – Scalable R Server functions
- Deploy/Consume – Using AzureML from R Server

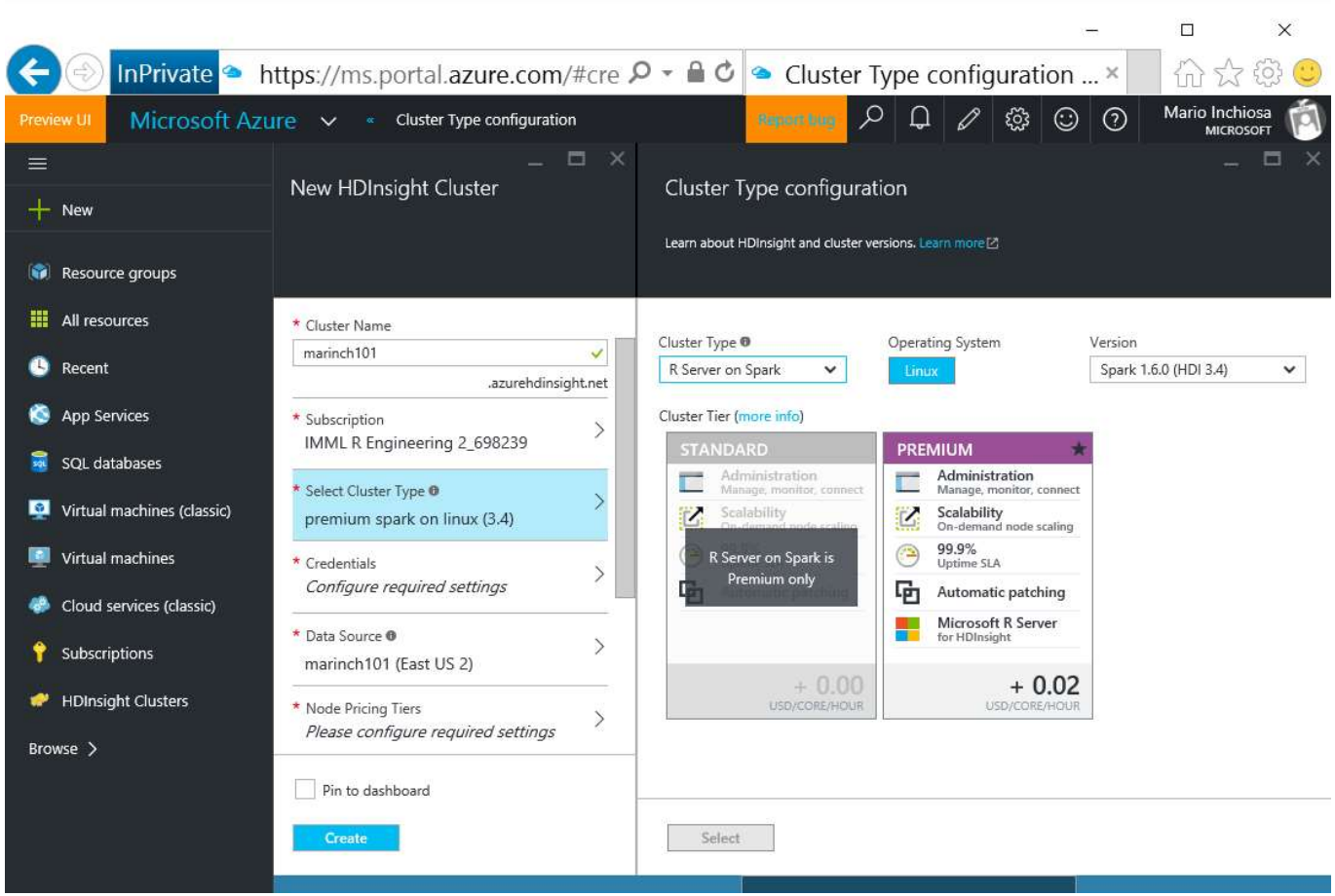
Airline data set

- Passenger flight on-time performance data from the US Department of Transportation's TranStats data collection
- >20 years of data
- 300+ Airports
- Every carrier, every commercial flight
- http://www.transtats.bts.gov

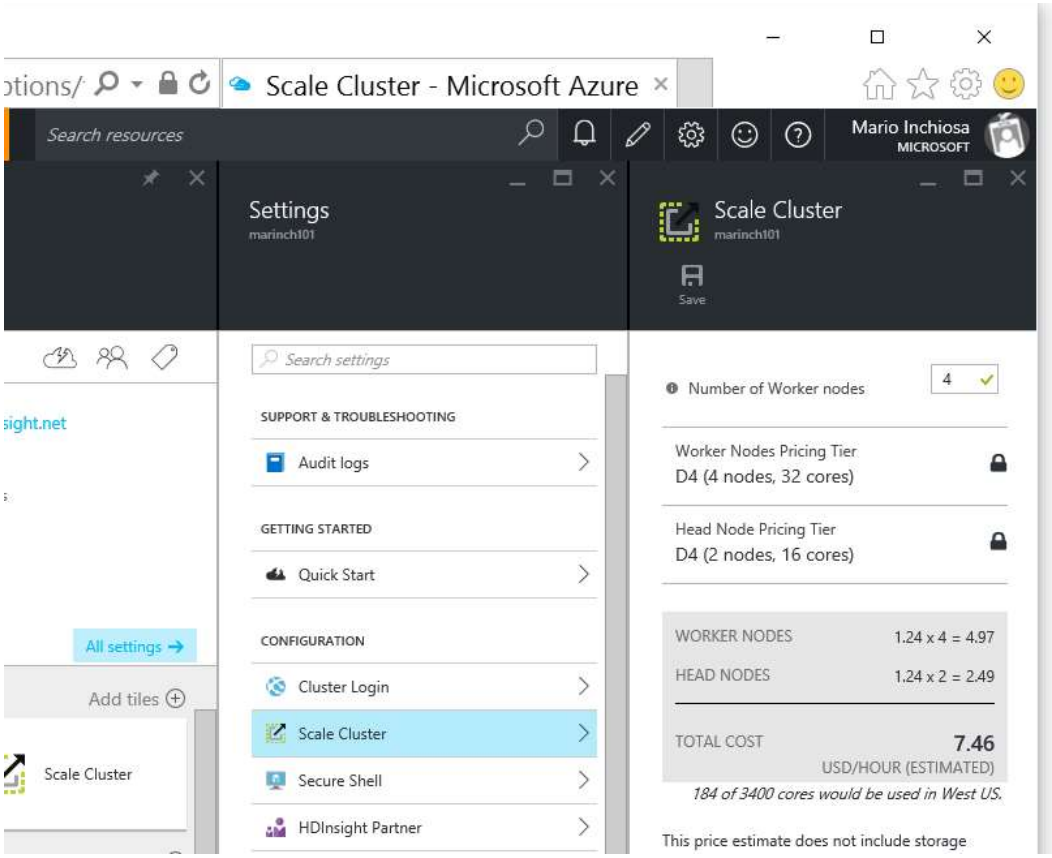
Weather data set

- Hourly land-based weather observations from NOAA
- > 2,000 weather stations
- http://www.ncdc.noaa.gov/orders/qclcd/

Provisioning a cluster with R Server



Scaling a cluster



Clean and Join using SparkR in R Server

```
#############################################################
# Join airline data with weather at Origin Airport
#############################################################

joinedDF <- SparkR::join(
  airDF,
  weatherDF,
  airDF$OriginAirportID == weatherDF$AirportID &
    airDF$Year == weatherDF$AdjustedYear &
    airDF$Month == weatherDF$AdjustedMonth &
    airDF$DayofMonth == weatherDF$AdjustedDay &
    airDF$CRSDepTime == weatherDF$AdjustedHour,
  joinType = "left_outer"
)
```

Train, Score, and Evaluate using R Server

```
##############################################
# Train and Test a Decision Tree model
##############################################

# Train using the scalable rxDTree function

dTreeModel <- rxDTree(formula, data = trainDS,
                      maxDepth = 6, pruneCp = "auto")

# Test using the scalable rxPredict function

rxPredict(dTreeModel, data = testDS, outData = treePredict,
          extraVarsToWrite = c("ArrDel15"), overwrite = TRUE)
```

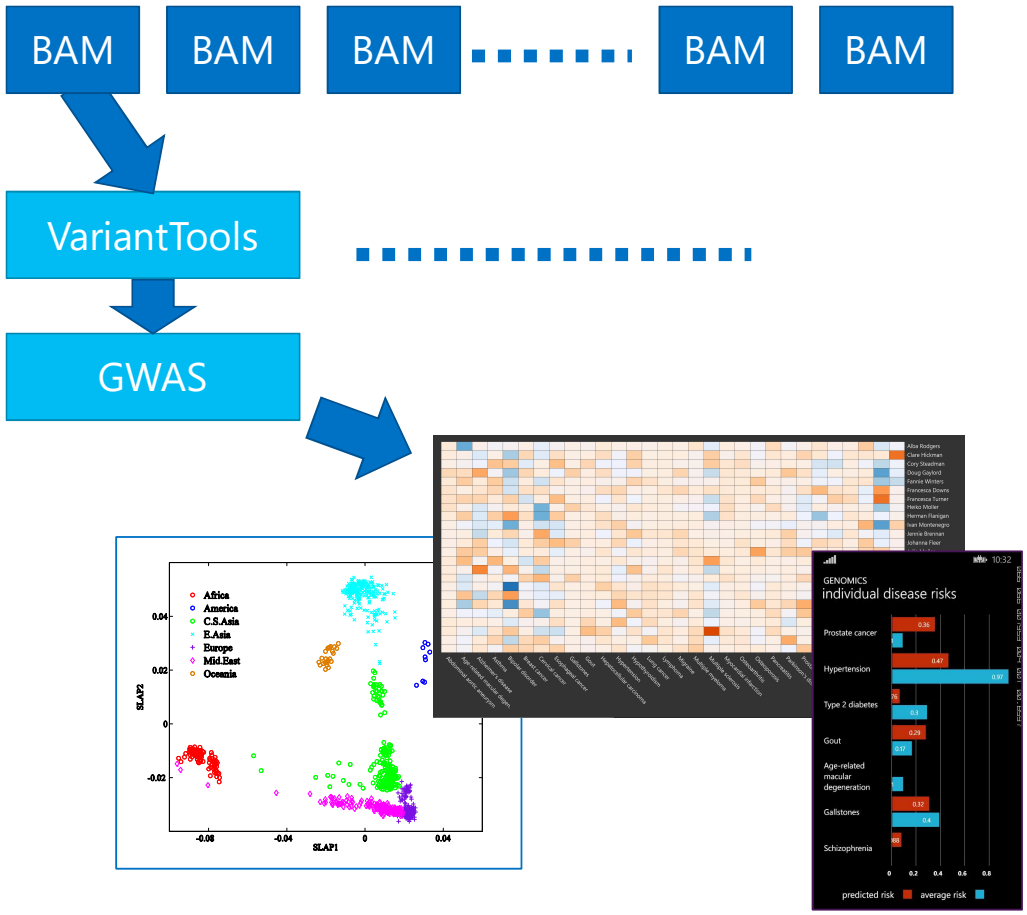
Publish Web Service from R

```
##################################################
# Publish the scoring function as a web service
##################################################

library(AzureML)

workspace <- workspace(config = "azureml-settings.json")

endpoint <- publishWebService(workspace, scoringFn,
                              name="Delay Prediction Service",
                              inputSchema = exampleDF)

##################################################
# Score new data via the web service
##################################################

scores <- consume(endpoint, dataToBeScored)
```

Demo Technologies

- HDInsight Premium Hadoop cluster
- Spark on YARN distributed computing
- R Server R interpreter
- SparkR data manipulation functions
- RevoScaleR Statistical & Machine Learning functions
- AzureML R package and Azure ML web service

Building a genetic disease risk application with R



Data

- Public genome data from 1000 Genomes
- About 2TB of raw data

Platform

- HDInsight Hadoop (8 clusters)
  - 1500 cores, 4 data centers
- Microsoft R Server

Processing

- VariantTools R package (Bioconductor)
- Match against NHGRI GWAS catalog

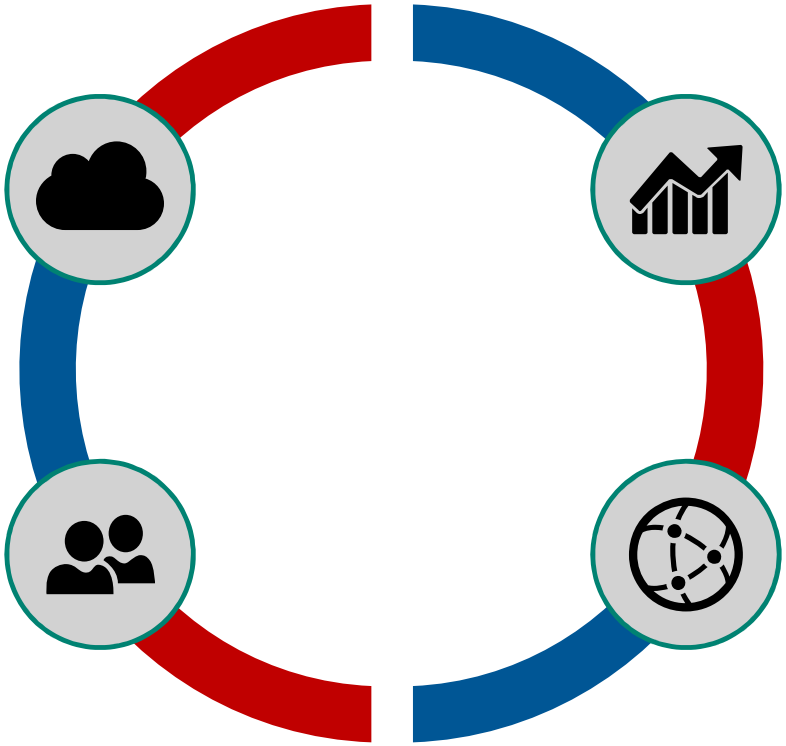
Analytics

- Disease Risk
- Ancestry

Presentation

- Expose as Web Service APIs
- Phone app, Web page, Enterprise applications

The Four Transformational Trends



cloud computing

2011 ➔ 2016 5x increase

big data

90% of the data in the world today has been created in the last two years alone

data science

Universities filling 300,000 US talent gap

open source

including R, Linux, Hadoop

For more information…

HDInsight Premium
microsoft.com/hdinsight

R Server
microsoft.com/r-server