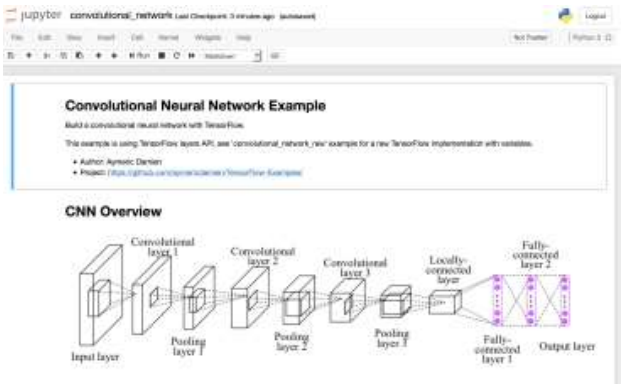
# Watson Machine Learning Accelerator

# Watson Machine Learn – Accelerator

- IBM PowerAI is a packaging of ML/DL frameworks for Linux on Power systems
  - Tensorflow, Caffe, Pytorch….
    - Compiled and optimized for IBM Power Systems
  - Growing number of frameworks since first release

- IBM WML-A is PowerAI + cluster management framework and deep learning platform:
  - IBM Spectrum Conductor and Deep Learning Impact
  - Notebooks, Docker, Distributed Deep Learning, Fabric algorithms

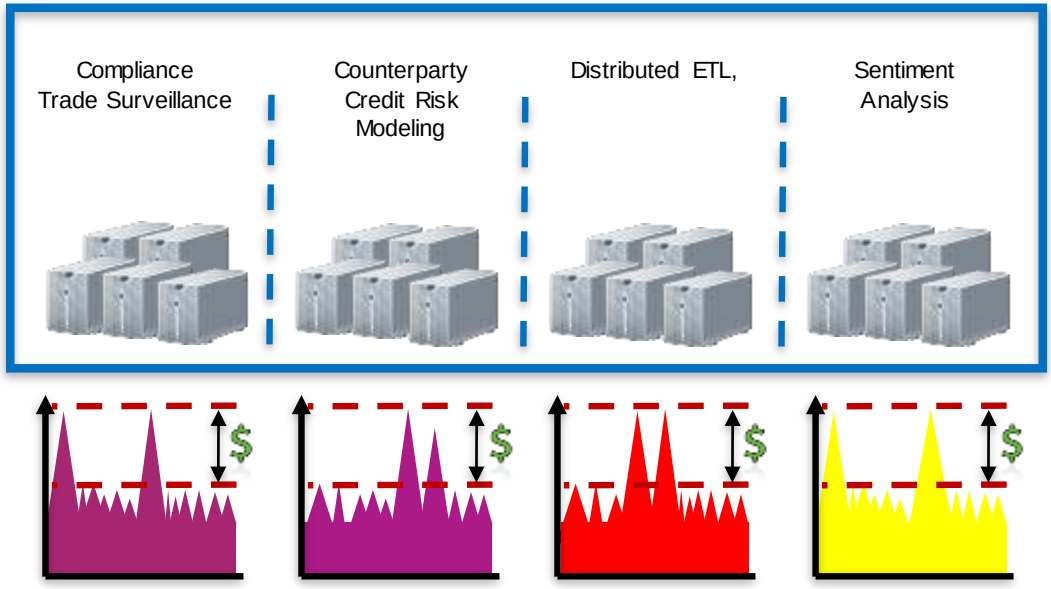
# Apache Spark


- Apache Spark is an open-source cluster-computing framework.
- Spark facilitates the implementation of iterative algorithms and exploratory data analysis.
- Spark schedules jobs through a cluster management system and requires a distributed filesystem.
- Why Spark?
  - Unified Analytics Platform
  - Multi-language (Python, Scala, R, SQL...)
  - Performance: faster than MapReduce
  - Diverse ecosystem
  - Very active open source project


# Challenges managing spark applications

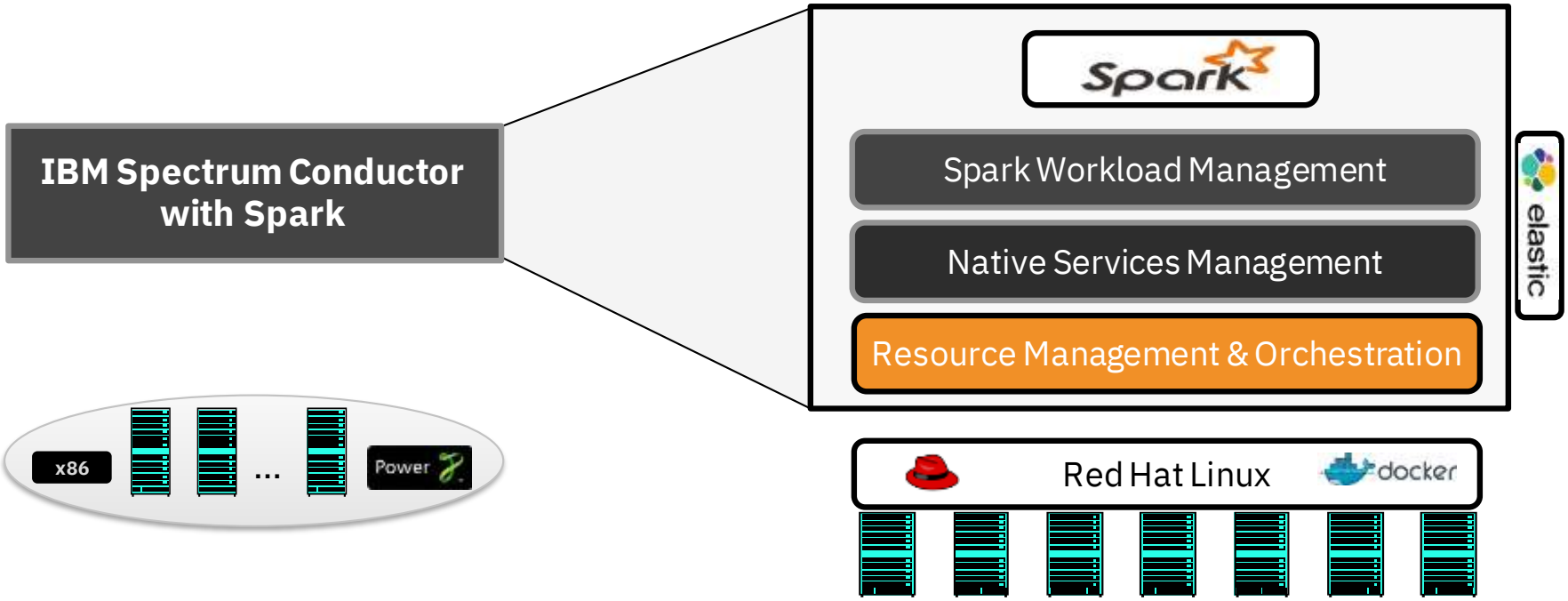
- In a word: siloed environments

- Different Lines of Business
- Multiple Spark versions
- Multiple notebooks and versions
- Security, governance
- SLAs
- Development, test, production
- Diverse data sources




Low utilization → Higher cost

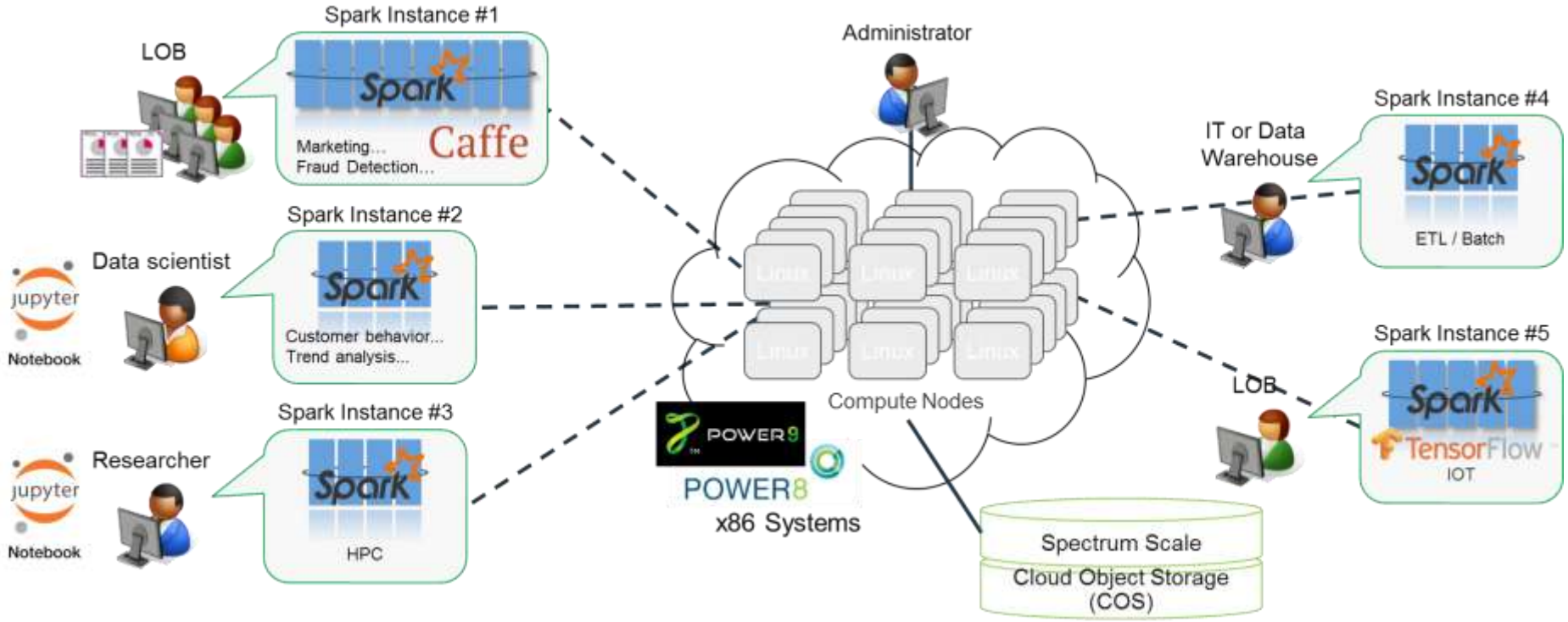
# Spectrum Conductor with Spark



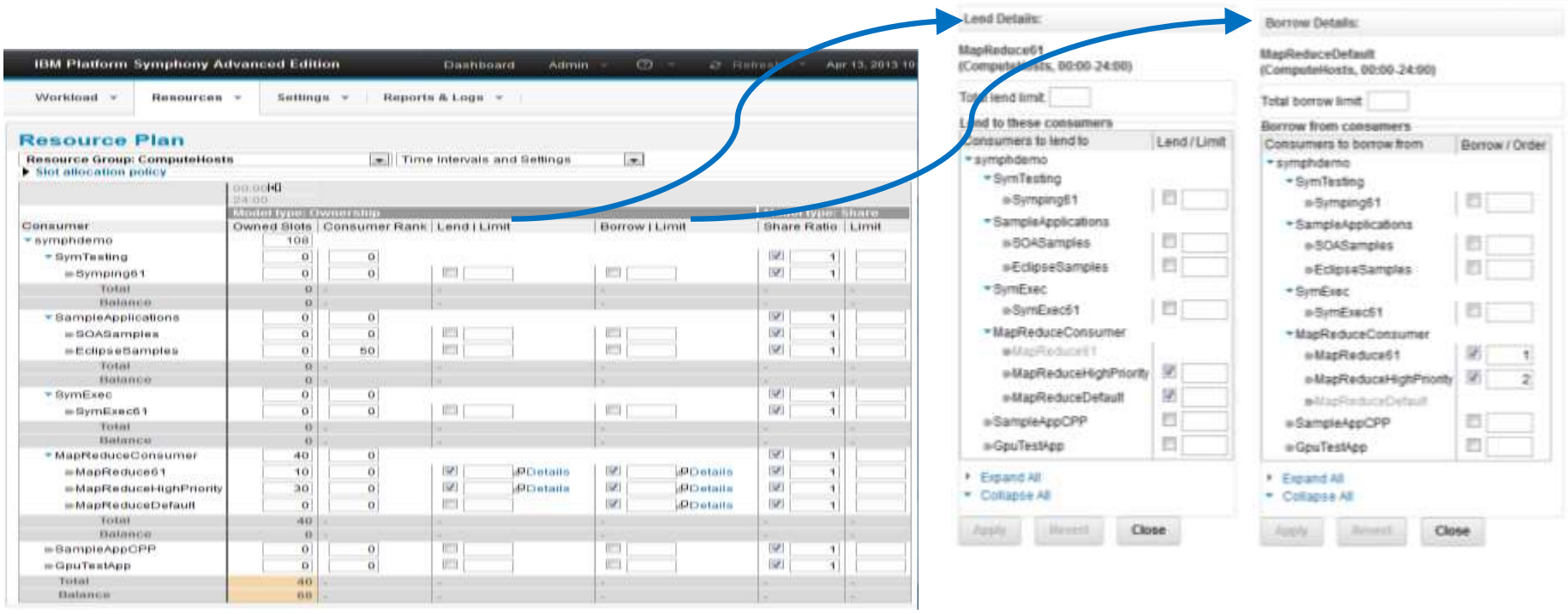
# Key concepts

- Instance groups
  - Defines a spark cluster
  - Introduces multi-tenancy
  - Isolates environments (security)
- Users and consumers
  - How binding is done at the OS level
  - Impersonation of a consumer
- Resource groups
  - Defines a pool of resources
    - » CPU resources
    - » GPU resources
  - Defines slots for resource management
- Resource plans
  - Sharing of resources
  - Reduced silos

# Instance groups

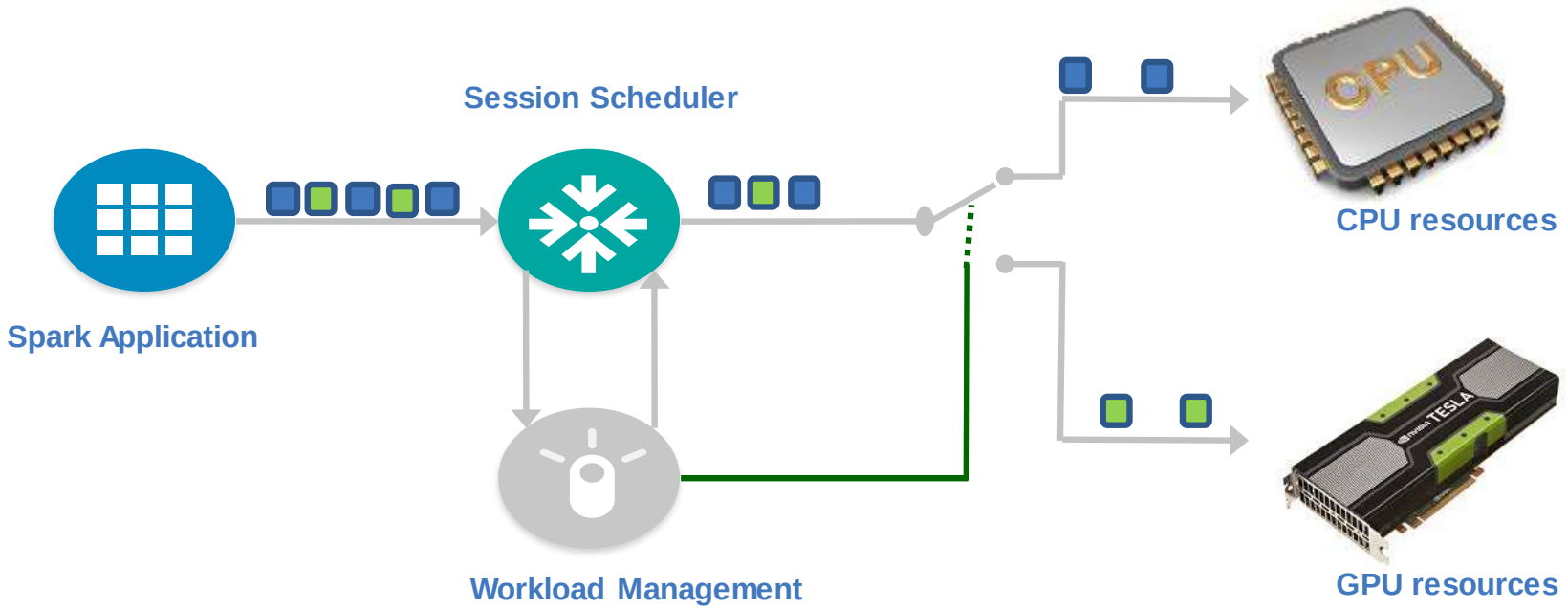


# Resource plans




- Sharing of resources while preserving ownership
- Change plan on-the-fly
- Allocations happen in runtime (dynamic allocation)
- Enables SLA management

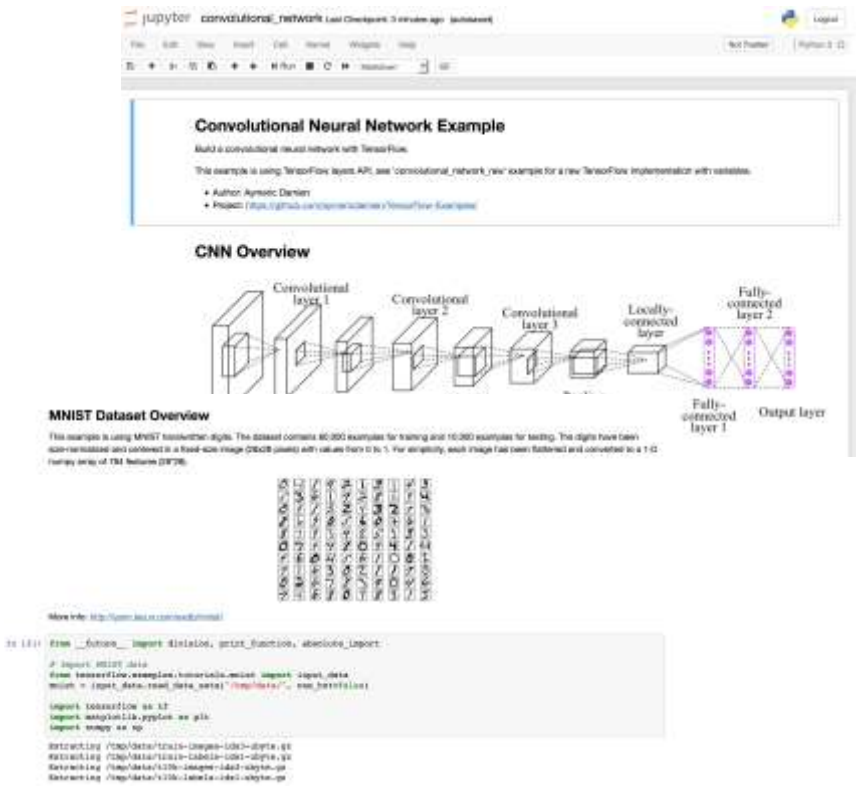
# GPU support




- Accelerating Spark applications with GPUs
  - Conductor scheduler interfaces with Spark scheduler to ensure that GPU resources are assigned to the applications that can use them.

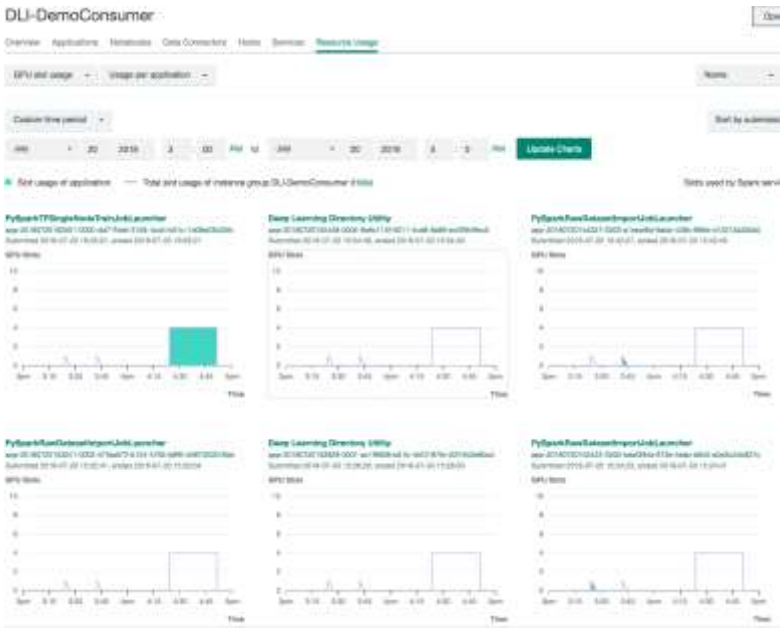
# Jupyter Notebooks | Docker

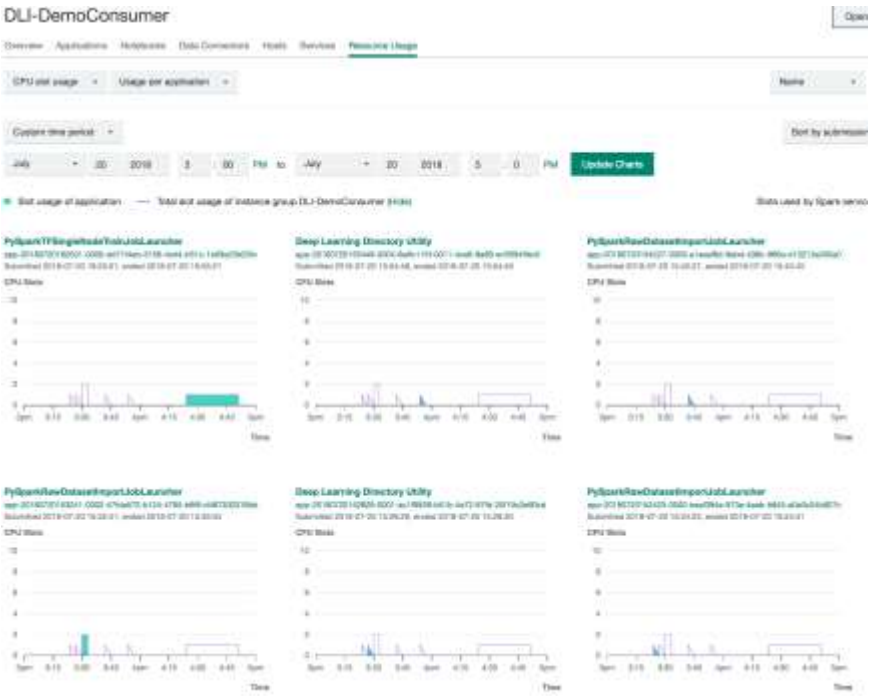
- Notebooks are created within an instance group
- Created for a user
- May leverage collaboration
- Fired off from Conductor




- Spectrum Conductor includes full integration with Docker
- Instance groups / notebooks may run in a Docker container

# Monitoring



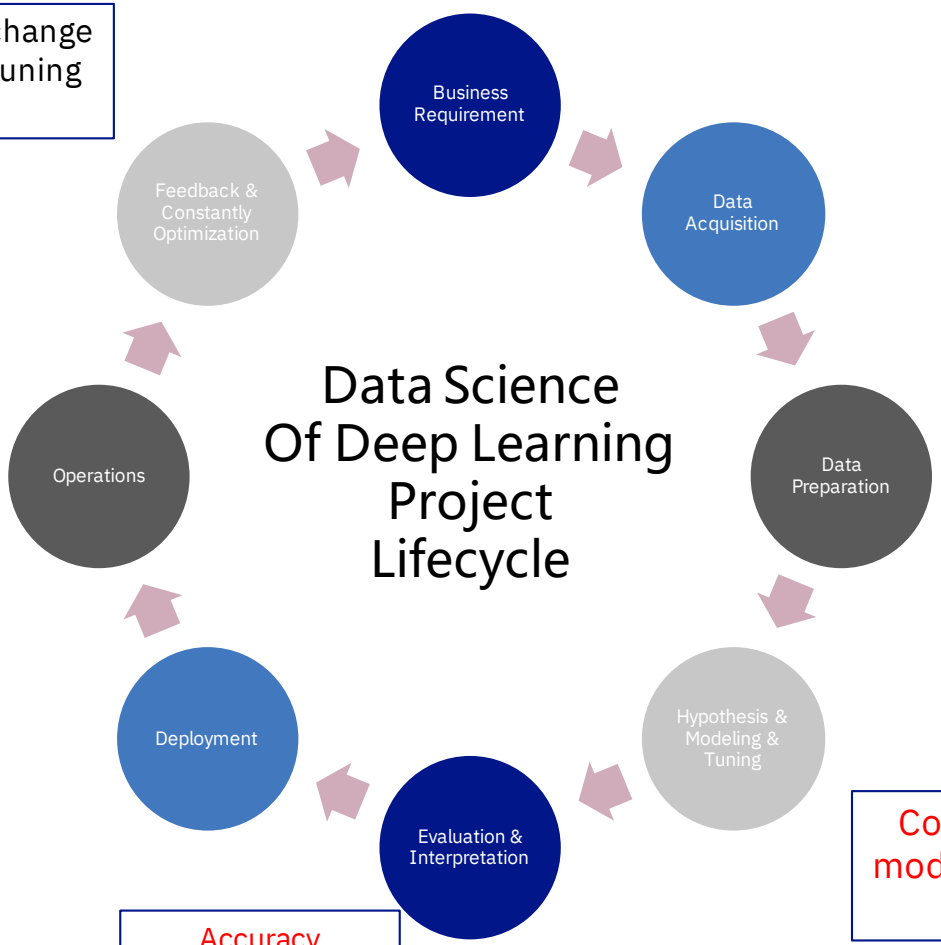




- Integrated Elastic Search, Logstash, Kibana for customizable monitoring
- Built-in monitoring Metrics
  - Cross Spark Instance Groups
  - Cross Spark Applications within Spark Instance Group
  - Within Spark Application
- Built-in monitoring inside Zeppelin Notebook

# Deep Learning Impact

# Challenges of deep learning

Business model / user data change → constant neural network tuning or training required




Most time is spent here ~80%

Unified AI platform Maximize resource utilization

Core piece. Understanding model issues, tuning models, long training runs

Accuracy
Overfitting
Underfitting
Hyper parameters

# Spectrum Conductor DLI

## Faster time to results

Distributed training on multiple servers and GPUs includes optimized software and frameworks to accelerate training times.

## Improve accuracy

Greater neural network model accuracy with hyper-parameter search and optimization, and with training visualization and tuning assistance.

## Reduce time preparing data

Less time spent importing, transforming and preparing data. Use Spark to manage data sources and imports.

## Improve ROI with shared resources

Better ROI with multi-tenant access to shared resources, which allow multiple data scientists to run different models at the same time on the same resources.

## Simplify administration

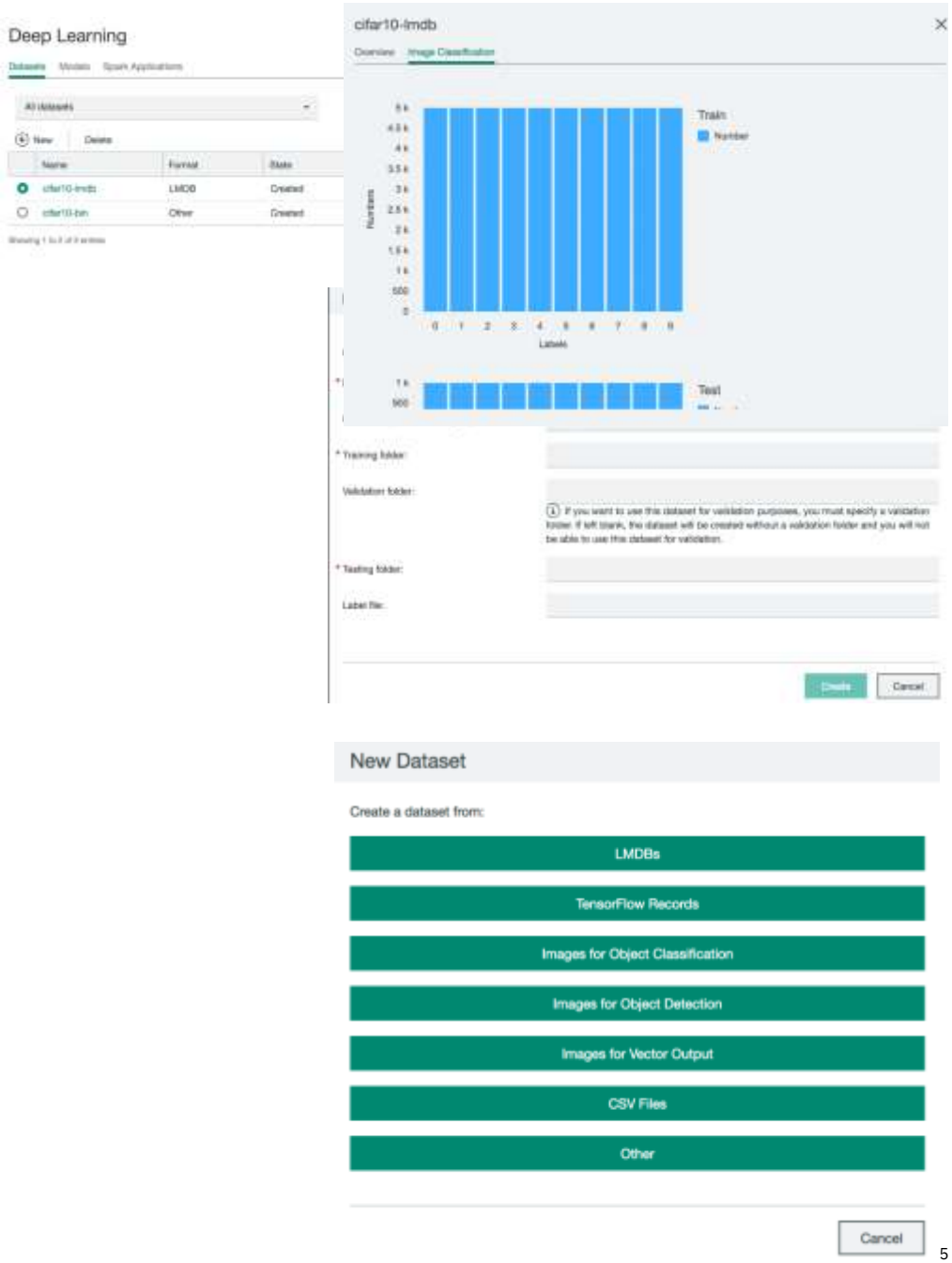
A consolidated framework for deep learning, monitoring and reporting enables you to achieve faster time to results with simplified management.

## Add to IBM Spectrum Conductor

Add a deep learning solution to IBM Spectrum Conductor. This highly available multitenant framework is designed to build a shared, enterprise-class Apache Spark environment.

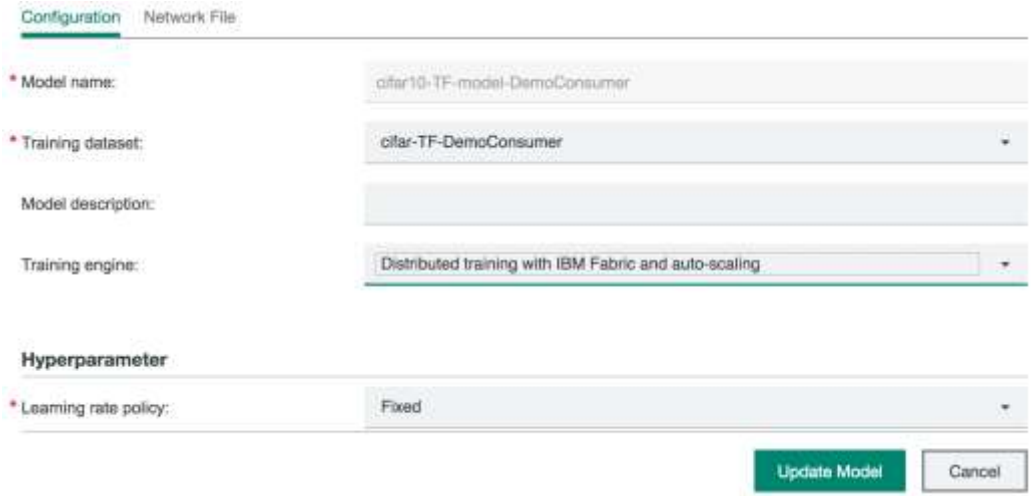
# Parallel data preparation

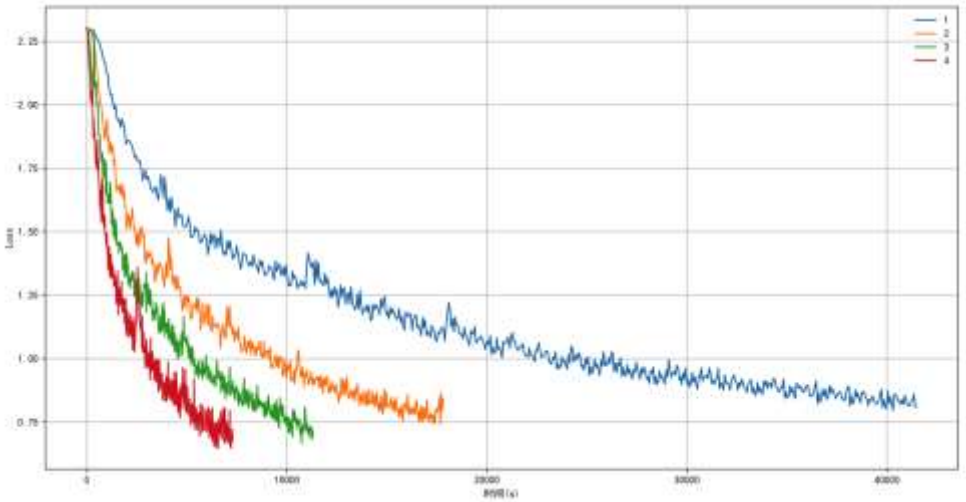
- Transform Data
  - Different Data dimension processing
  - Resize data to fit the network input layer
- Algorithm to keep the distribution of data
  - Rescaling by cross-entropy loss method
  - Hold-out vs Cross validation vs Bootstrapping
- Parallel Data Import
  - Integrate with ETL
  - Parallel transfer huge raw data to lmdb or tensor record format



# Parallel training

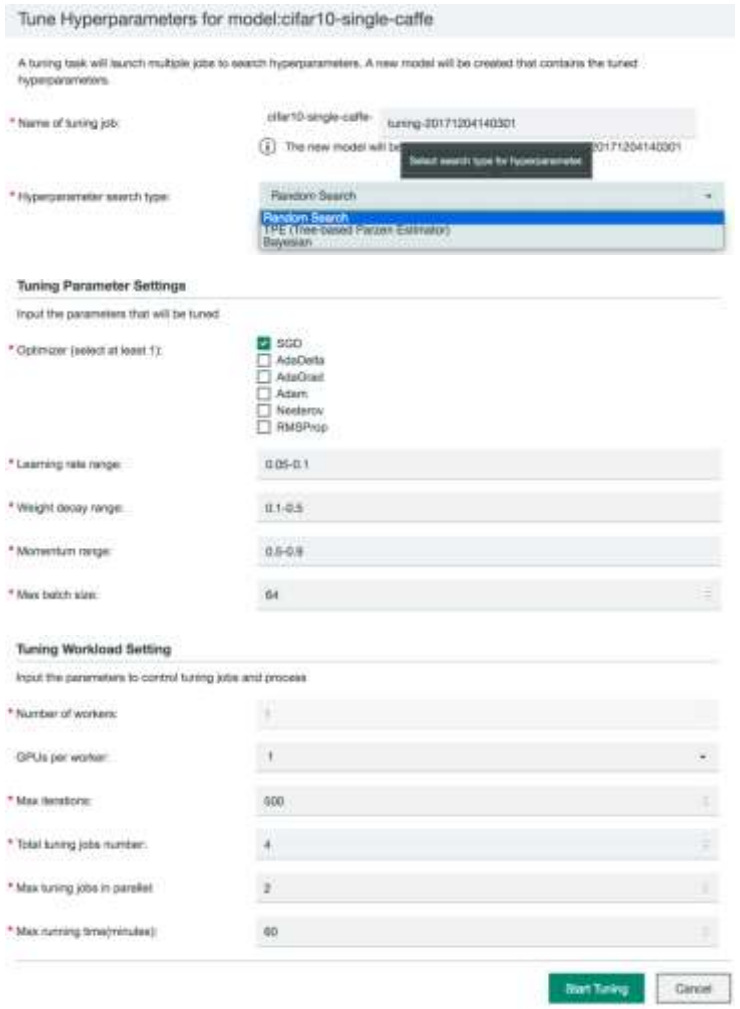
- Different optimizers in parallel
- Relationship among:
  - Iteration Number(τ)
  - Node number(K)
  - GPU Number(n)
  - Communication Overhead(s)
  - Accuracy
    - Ma(b,K,n,τ) vs single node

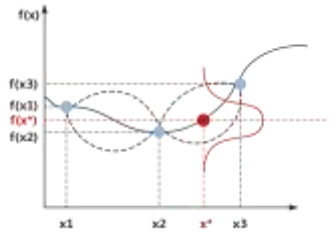


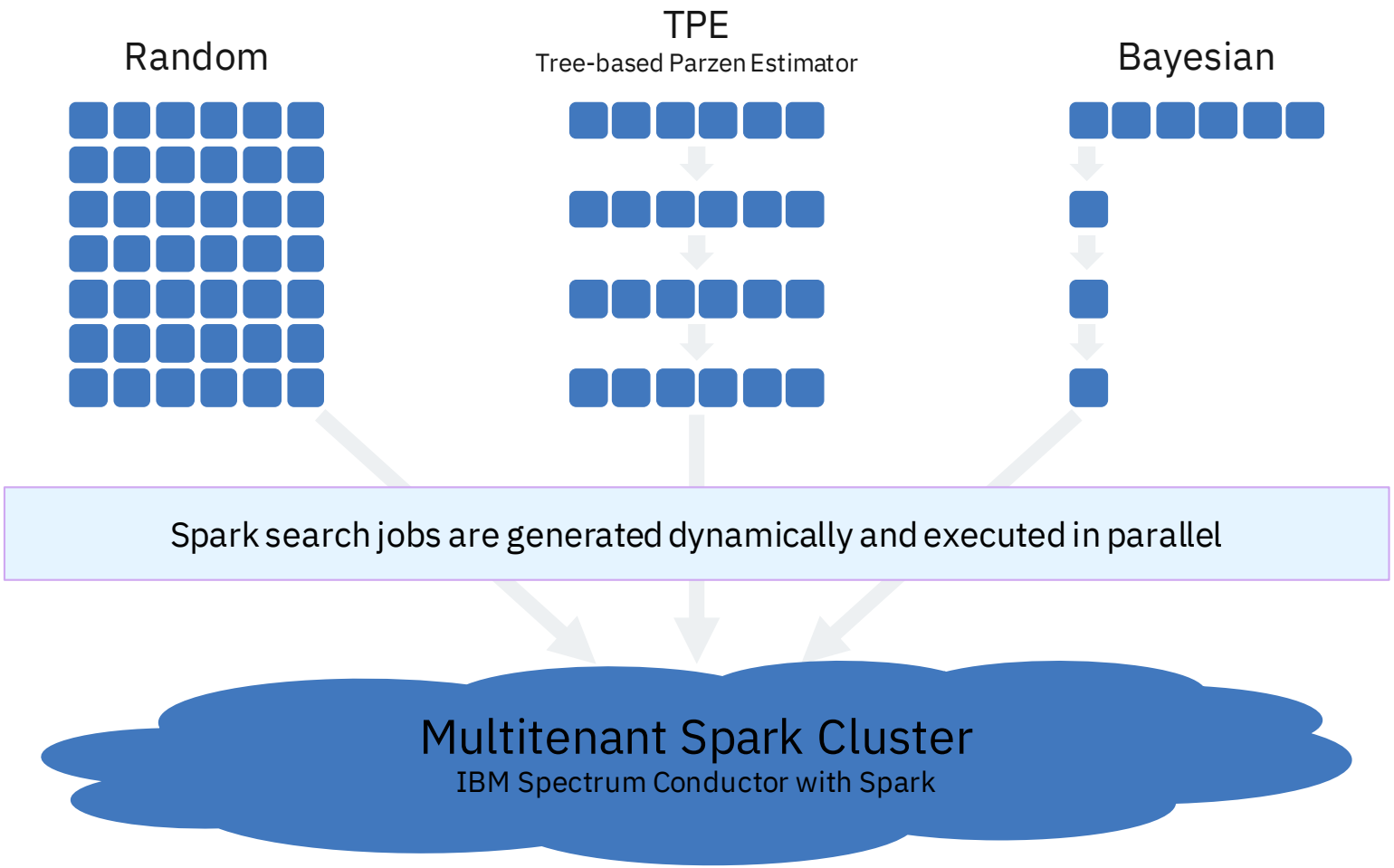
# Hyper parameters

- Search:
  - Random
    - The optimal solutions is above 5% in the whole space
    - 600 – 800 search may find a solution near the optimal solutions
  - Tree-structured Parzen Estimator
    - Modeled by generative process of hyper-parameters, replacing the distributions of the configuration prior with non-parametric densities
    - 10% additional calculation effort than random with around 30% accuracy improvement
  - Bayesian Estimator
    - Widely sample data and leverage multivariate Gaussian distribution get the θ
    - Calculate EI and choose new sample point
    - Bayesian provide better method than TPE to jump out a local optimal solution
    - Better accuracy with massive trained result
- Parameter setting: optimizer, learning rate, weight decay, momentum.
- Workload setting: # of workers and GPUs, iterations, and so on.



# Hyper parameter search

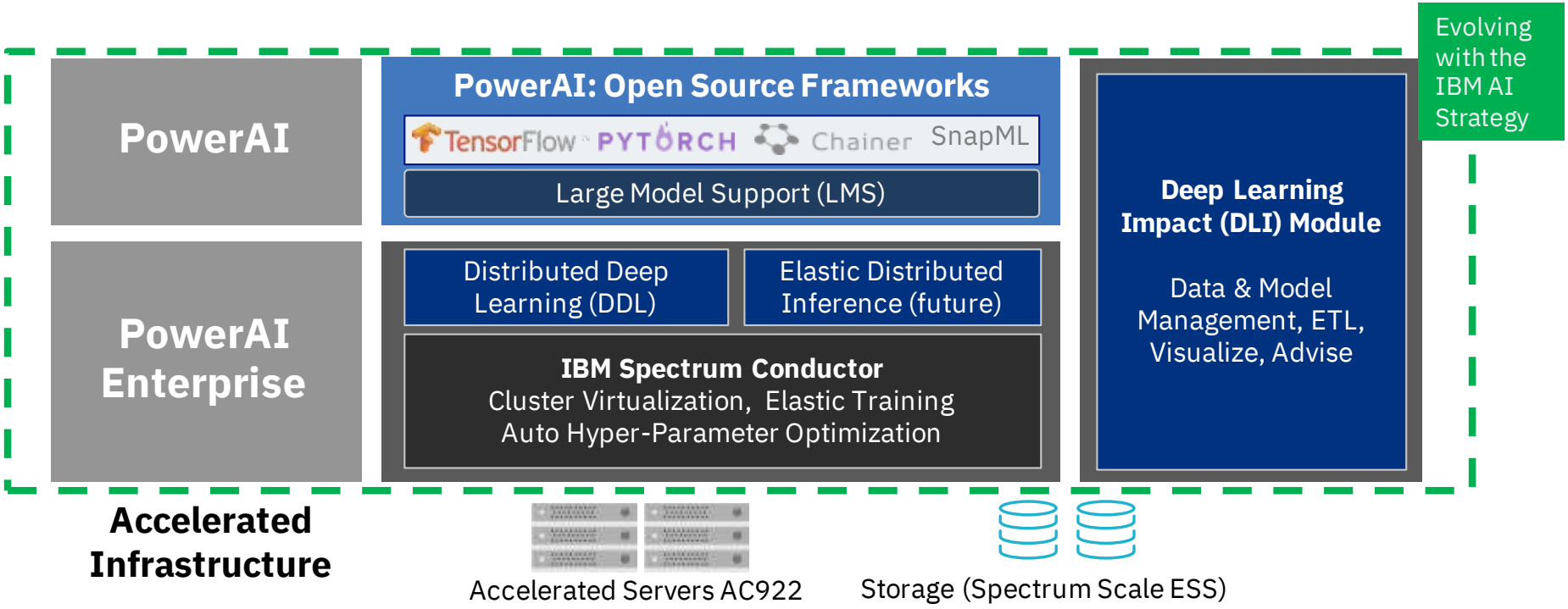


# Monitoring, Advisor, Optimizer

- Neural network has the property of **long time to train**, **easy to cause exception** and **communication overhead** when considering distributed DL service.
- Neural network takes long time to search a good combination of **hyper-parameters**, the consumption time will be **exponential increase** with the size of hyper-parameters and its range.
- Neural network is so complex that it is hard for users to build an **end to end solution** including determining **performance metrics**, choosing the **baseline models**, deciding whether to **gather more data**, when to **early stop**, and selecting **hyper-parameters**.
  - DLI can detect issues:
    - Gradient Explosion
    - Overflow
    - Saturation
    - Divergence
    - Overfitting
    - Under fitting
  - And suggest parameter tuning!

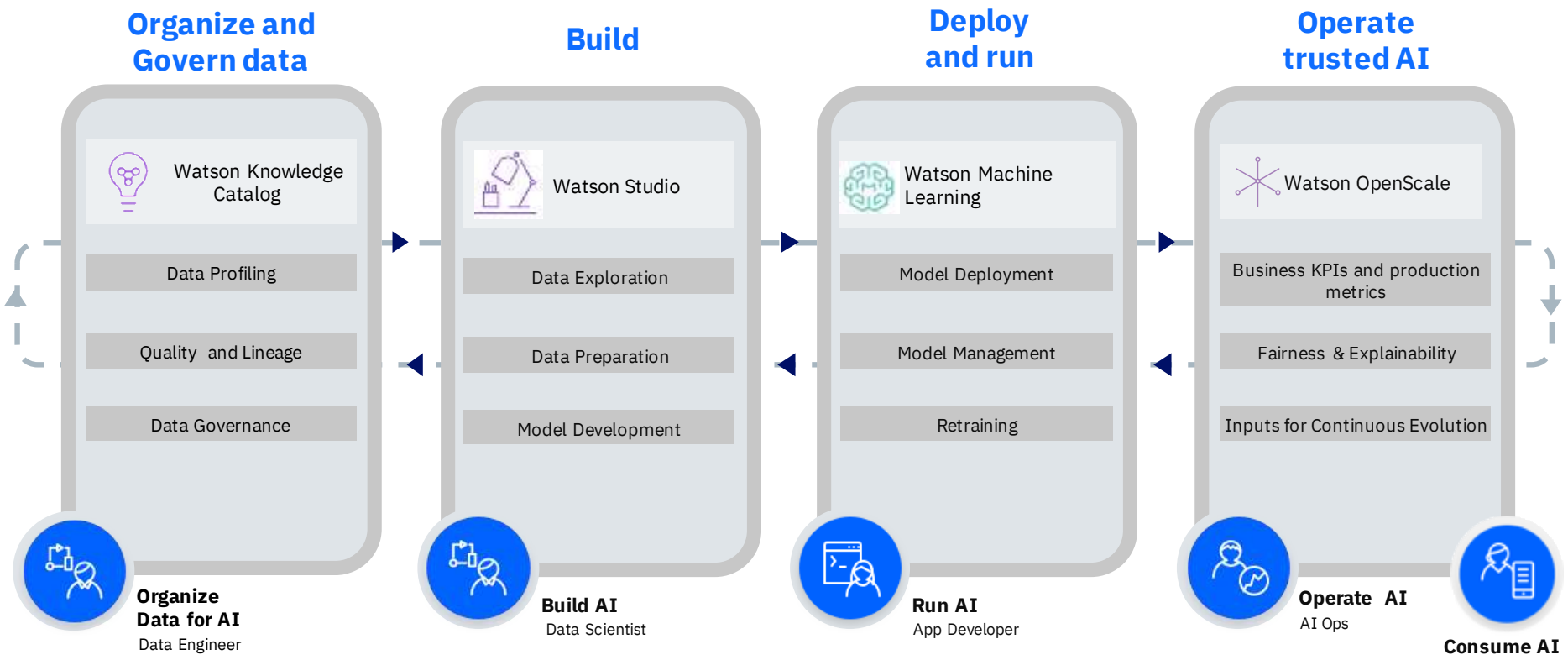
# Recent announcements

# Recent announcements @ THINK




PowerAI Enterprise → Watson Machine Learning – Accelerator Integration with Watson suite

# Recent announcements @ THINK

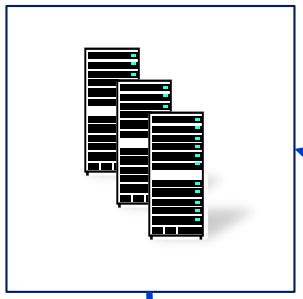


# Recent announcements @ THINK

**Watson Studio**

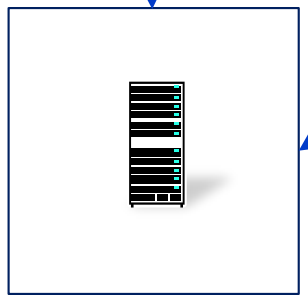
**IDE and Notebooks**

**Watson Machine Learning**



**Offload:**
- Spark and Notebook execution via Jupyter gateway

Model Management and Execution



**Offload:**
- Model Training and HPO tuning
- API submission and integration
- Distributed Inference

WML – Accelerator
Cluster of servers

**Deployment:**
- Add multi-node AC922 cluster for AI distributed training and execution
- GPU scheduling and management across Studio/WML workloads
- Support bare metal or ICP deployment

AI Starter Kit
- 2 x AC922
- 1 x LC922 with
- WML-A software
- Simplified configuration, ordering and fulfillment