Minsky Cluster Training

—

Eric Fiala
Cognitive Solution Architect

IBM

Agenda

Cluster Basics
Cluster Overview
Cluster Access

Spark
Spark Instance Group Overview
Submitting Basic Spark Jobs

Deep Learning
Deep Learning Overview
Importing Datasets
Running Basic Training Jobs
Hyperparameter Tuning
Elastic Distributed Training

Jupyter / Anaconda
Jupyter / Anaconda overview
Work with anaconda through notebook
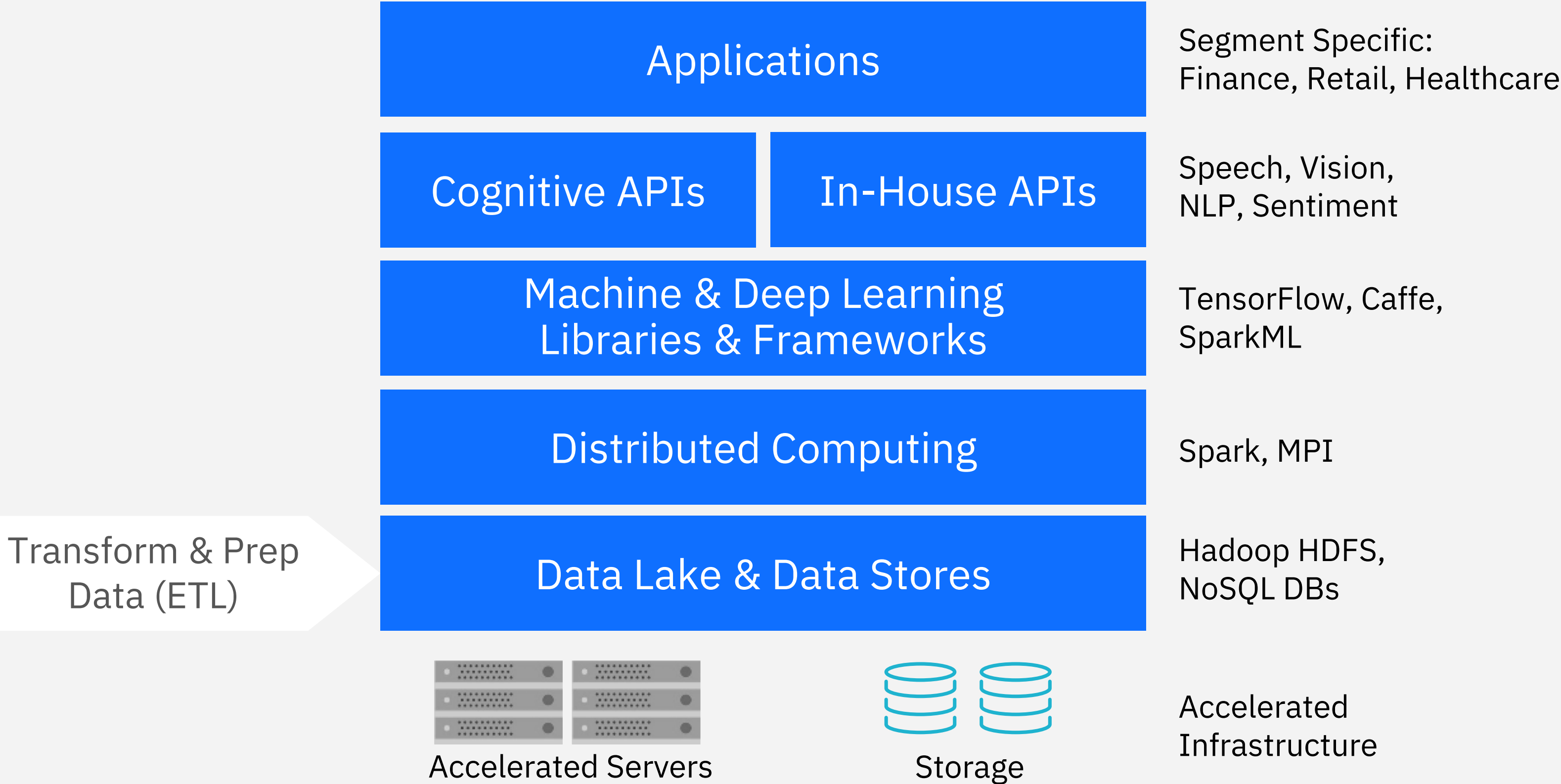Work with anaconda through CLI

Open Discussion / Questions

Conclude

Cluster Basics

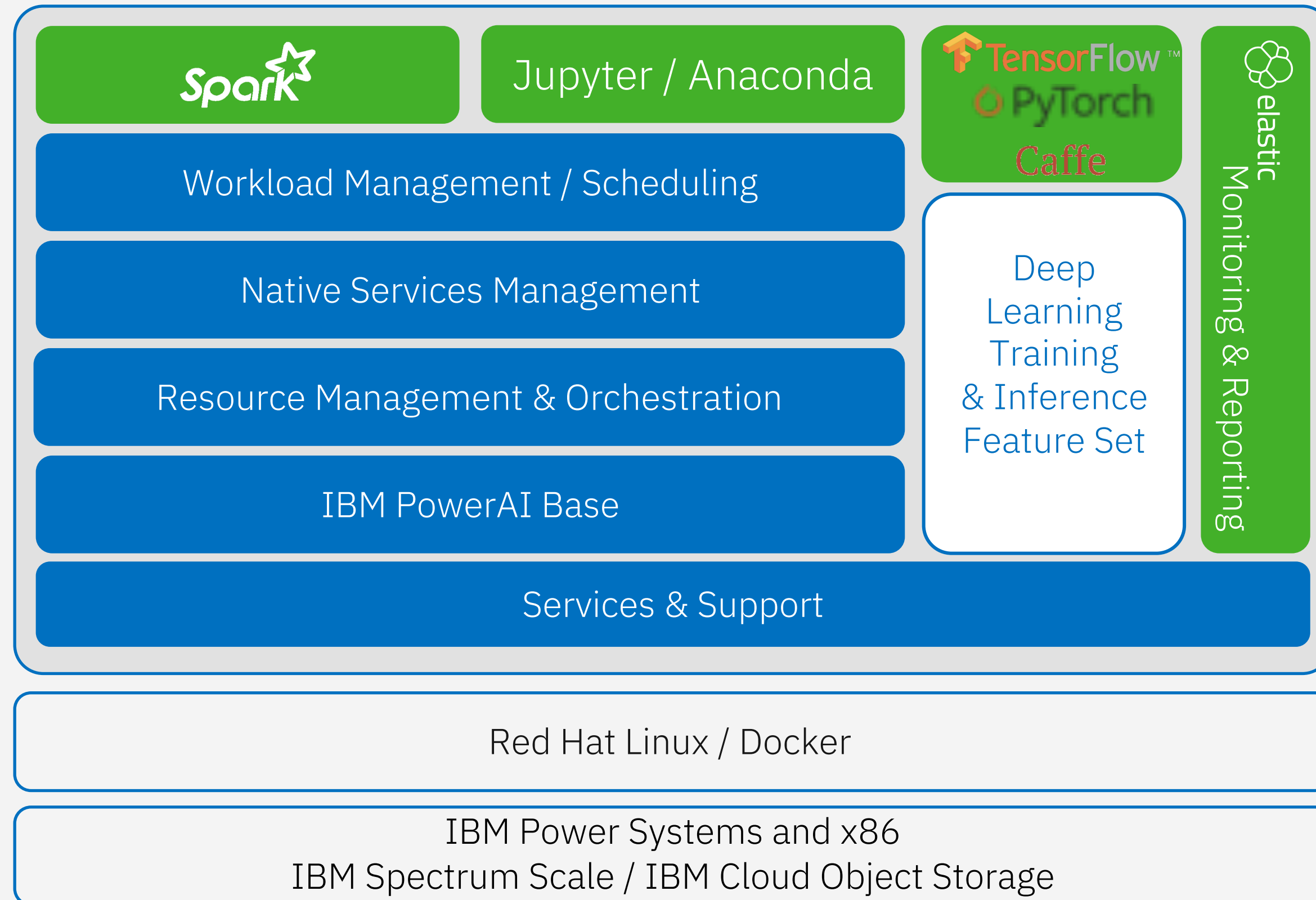# It's a Process... and it all needs to work together in order to work correctly!

Organizations generate a ton of data
Data that requires governance

Which must be cleaned and shaped for training

...then models must be designed

...that must be hosted and monitored

To select an optimal model

... and trained on high performance compute infrastructure

AI Infrastructure Stack

Applications — Segment Specific: Finance, Retail, Healthcare

Cognitive APIs | **In-House APIs** — Speech, Vision, NLP, Sentiment

Machine & Deep Learning Libraries & Frameworks — TensorFlow, Caffe, SparkML

Distributed Computing — Spark, MPI

Data Lake & Data Stores — Hadoop HDFS, NoSQL DBs

Transform & Prep Data (ETL)

Accelerated Servers

Storage

Accelerated Infrastructure

5

IBM Watson ML Accelerator:
Power AI Base + Spectrum Conductor + Deep Learning Impact

Spark | **Jupyter / Anaconda** | **TensorFlow™ PyTorch Caffe** | **elastic Monitoring & Reporting**

Workload Management / Scheduling

Native Services Management

Resource Management & Orchestration

IBM PowerAI Base

Deep Learning Training & Inference Feature Set

Services & Support

Red Hat Linux / Docker

IBM Power Systems and x86
IBM Spectrum Scale / IBM Cloud Object Storage

Multitenancy at Scale – Shared Services Architecture

- **Physical view** Single cluster of resources managed by IBM **Watson Machine Learning Accelerator**
- **Logical view** Each group has **their own cluster resources –** Isolated & Secured with SLA Managed Resource Allocation
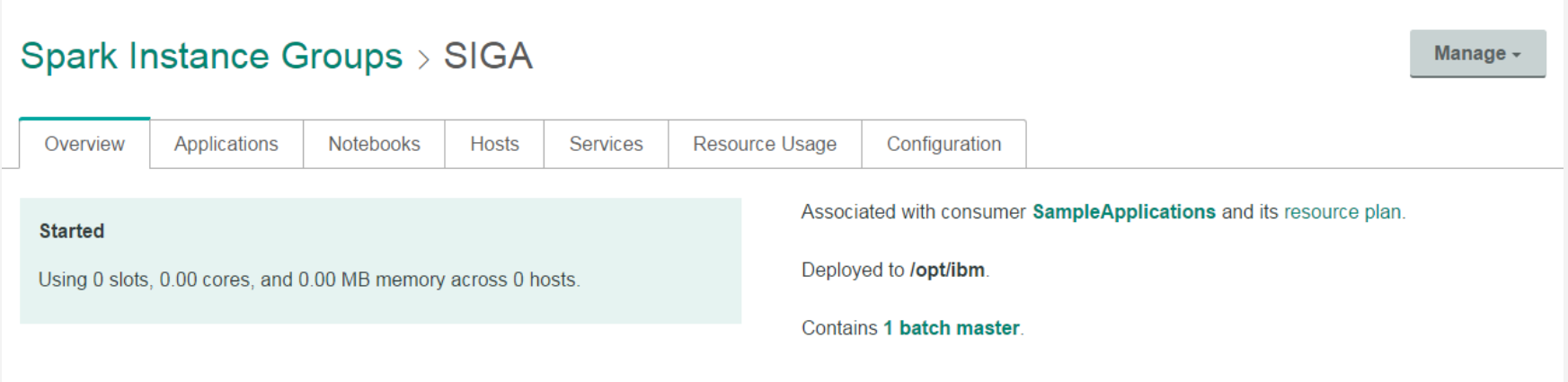
Cluster Access

- ## URL http://ibmminsky-head.cern.ch:8080/platform
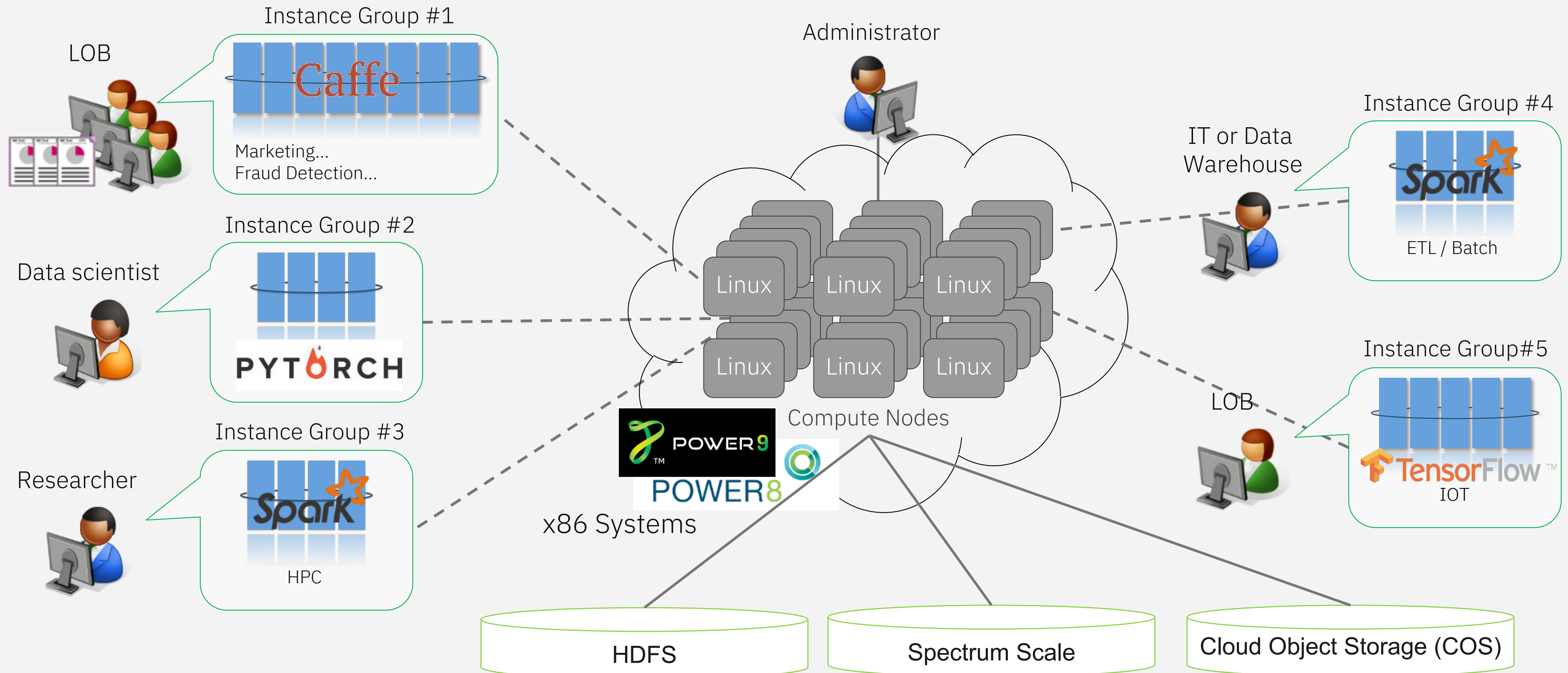
  - Use the fully qualified hostname (ibmminsky-head.cern.ch)

- ## User: your cern.ch username

  - Your username should be added to the appropriate cluster group by Admin team

- ## Password: your cern.ch password

Spark
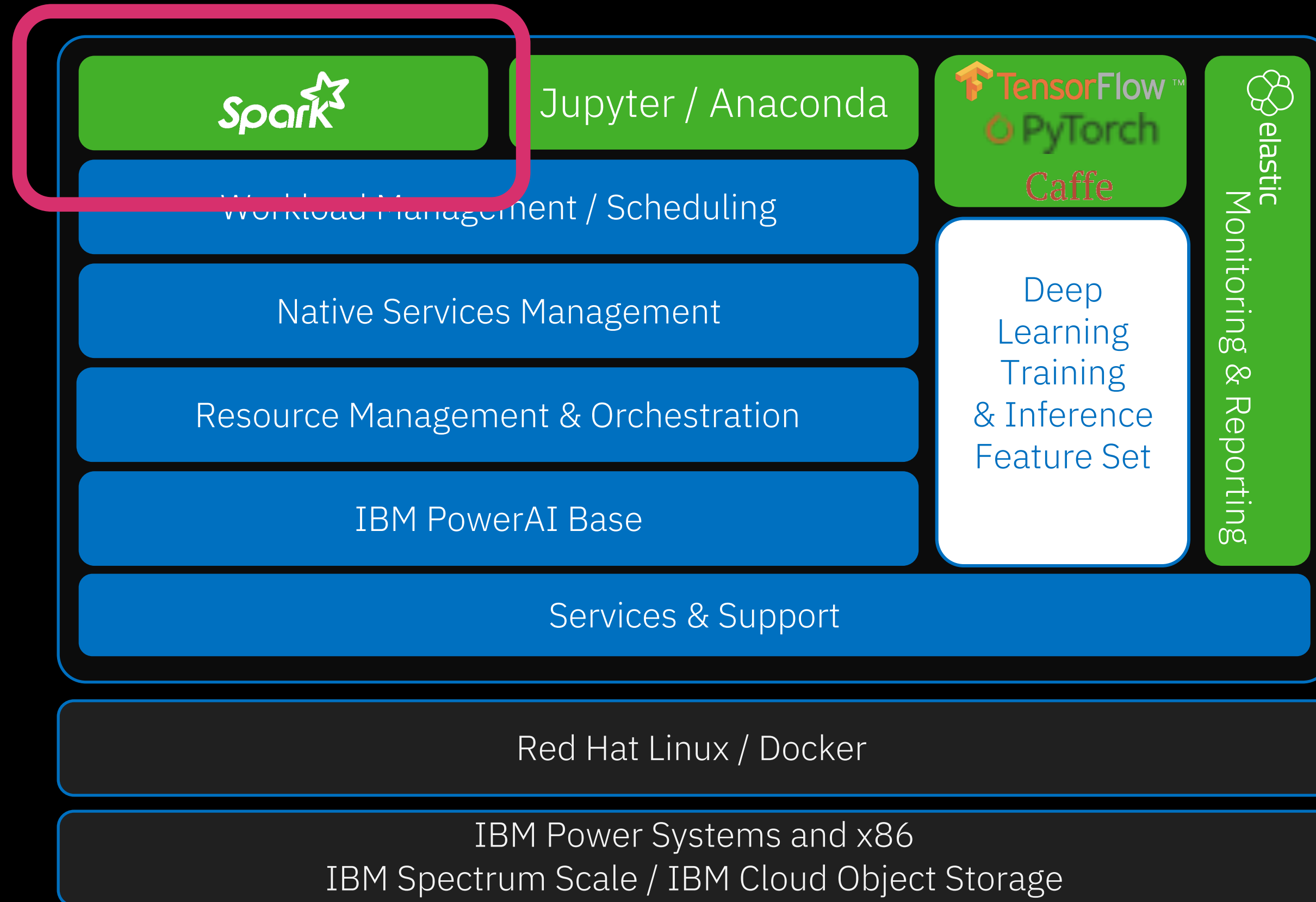
Spark Instance Group Overview

- A Spark instance group is an installation of Apache Spark that can be scaled to run multiple masters, with history server, shuffle service, and notebooks configured.

- Each Spark instance group is a collection of services that is associated with a top-level consumer.

- You can manage and monitor a Spark instance group, and drill down to manage the related services and service instances.

- A Spark instance group can be created to serve a line of business or a group member of a business organization.

- Multiple versions supported on the same cluster – (Spark 1.5.2, 1.6.0, 1.6.1, 2.0.1, 2.3.1, …)

Spark Instance Groups › SIGA

| Overview | Applications | Notebooks | Hosts | Services | Resource Usage | Configuration |

Started

Using 0 slots, 0.00 cores, and 0.00 MB memory across 0 hosts.

Associated with consumer **SampleApplications** and its resource plan.

Deployed to **/opt/ibm**.

Contains **1 batch master**.

Spark Instance Group Overview

- Runtime Isolation:

    - Driver/executor process are owned by submitter

    - Within same cluster, users can deploy multiple Spark versions simultaneously on the same resources

- Data at Rest Isolation:

    - Data/log can only be accessed by owner

- Management Isolation:

    - RBAC authorization

Submitting Basic Spark Jobs

- Submitting workload via Cluster management console

  - Workload > Spark > Spark Instance Group > Submit Application

- Programmatically Using REST API

More Info:

Submitting batch job
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/tutorial/t_submit_app.html

Monitoring batch jobs
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/managing_applications/applications_monitoring_all.html

RESTful API
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/get_started/locating_rest_apis.html

YouTube Playlist
https://www.youtube.com/playlist?list=PLdgu63Noj6vLSTPPRdD7uuP9Zej_9stsO

Submitting Basic Spark Jobs

- Using spark-submit command from CLI

  ```
  . /nfs/software/wmla/profile.platform
  egosh user logon
  cd spark-home-directory
  bin/spark-submit --master spark://master-url:master-port --deploy-mode cluster --class
  org.apache.spark.examples.SparkPi ./examples/jars/spark-examples_2.11-2.3.1.jar 100
  ```

- Find the master_url, port, and spark home directory in webui
  (Workload > Spark > Spark Instance Group > Overview)

Spark... Even More

Shared RDDs / Dataframes

- Persistent Resilient Distributed Datasets (RDDs)

- Load once, read many, in native spark format

Adaptive Scheduling

- Run spark tasks on CPU and/or GPU based on scheduling logic

More Info:

Developing shared RDD applications
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/managing_applications/applications_shared_develop.html

GPU RDDs
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/developing_instances/instances_gpu_run_app_gpurdd.html

GPU RDD example
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/reference_s/gpu_rdd_samples.html

Guidelines for application code
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/performance_tuning/application_code_considerations.html

Adaptive Scheduling
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/developing_instances/instances_gpu_adaptive_scheduling.html

Deep Learning

Spark	Jupyter / Anaconda	TensorFlow™ PyTorch Caffe	elastic Monitoring & Reporting
Workload Management / Scheduling		Deep Learning Training & Inference Feature Set	
Native Services Management			
Resource Management & Orchestration			
IBM PowerAI Base			
Services & Support			

Red Hat Linux / Docker

IBM Power Systems and x86
IBM Spectrum Scale / IBM Cloud Object Storage
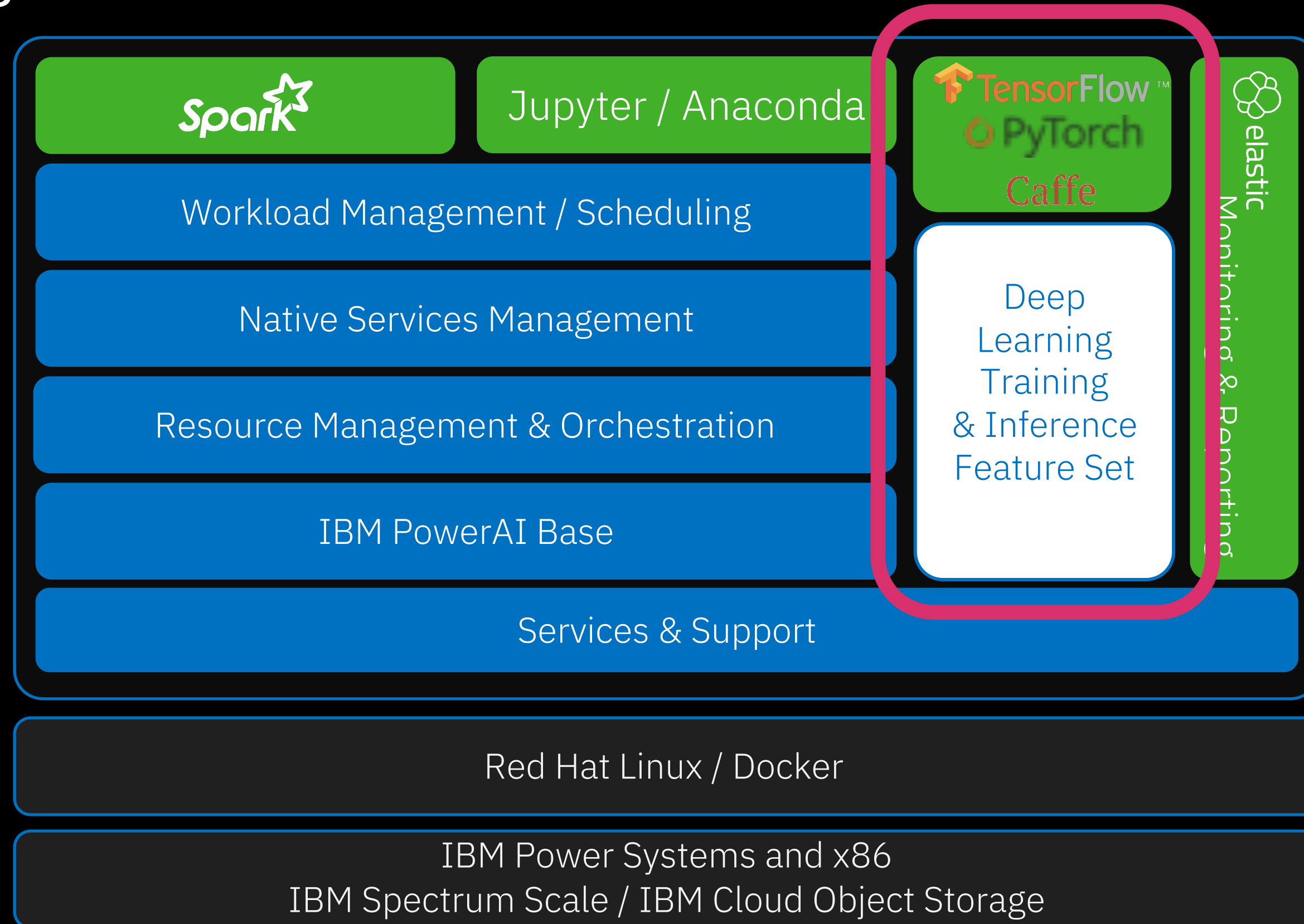
Basic Workflow for ML/DL Training & Execution

New Data

Data Source

- Traditional Business Data
- Sensor Data
- Data from collaboration partners
- Data from mobile apps & social media
- Legacy Data

Data Preparation

Data Ingestion

Heavy IO

Pre-Processing

Training Dataset

Testing Dataset

Model Training

Parallel Hyper-Parameter Search & Optimization

Monitor & Advise

Network Models

Hyper-Parameters

Iterate

Instrumentation

AI Deep Learning Frameworks (Tensorflow, Caffe, …)

Distributed & Elastic Training for Deep Learning

Inference

Deploy in Production using Trained Model (Rest API)

Trained Model Life Cycle Management

Importing Datasets

- Create different types of datasets

  - LMDB

  - TFRecords

  - Images for object classification

  - Images for object detection

  - Images for vector output

  - CSV files

  - Others (COPY, NLP (NER, POS, Segmentation), Text classification)

More Info:

https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/deep-learning-create-dataset.html

Running Basic Training Jobs

- Importing a model from model zoo

- Creating a new model from template

- Launch a training job

- Launch hyperparameter tuning

More Info:

Create a training model
https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/deep-learning-create-training-model.html

Edit models for training:
https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/configure-caffe-model.html
https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/configure-tensorflow-model.html
https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/configure-pytorch-model.html

Model zoo:
https://us-south.git.cloud.ibm.com/ibmconductor-deep-learning-impact

End to end workflow:
https://developer.ibm.com/tutorials/use-computer-vision-with-dli-watson-machine-learning-accelerator/

Hyperparameter Tuning

- Find optimal hyperparameters faster

  - Search algorithms (random, TPE, Bayesian, Hyperband)

  - Parallel search maximizing resource use

  - Option for custom parameters

More Info:

Edit model for hyperparameter tuning
https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/configure-tensorflow-model-hyperparameter-tuning.html
https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/configure-pytorch-model-hyperparameter-tuning.html
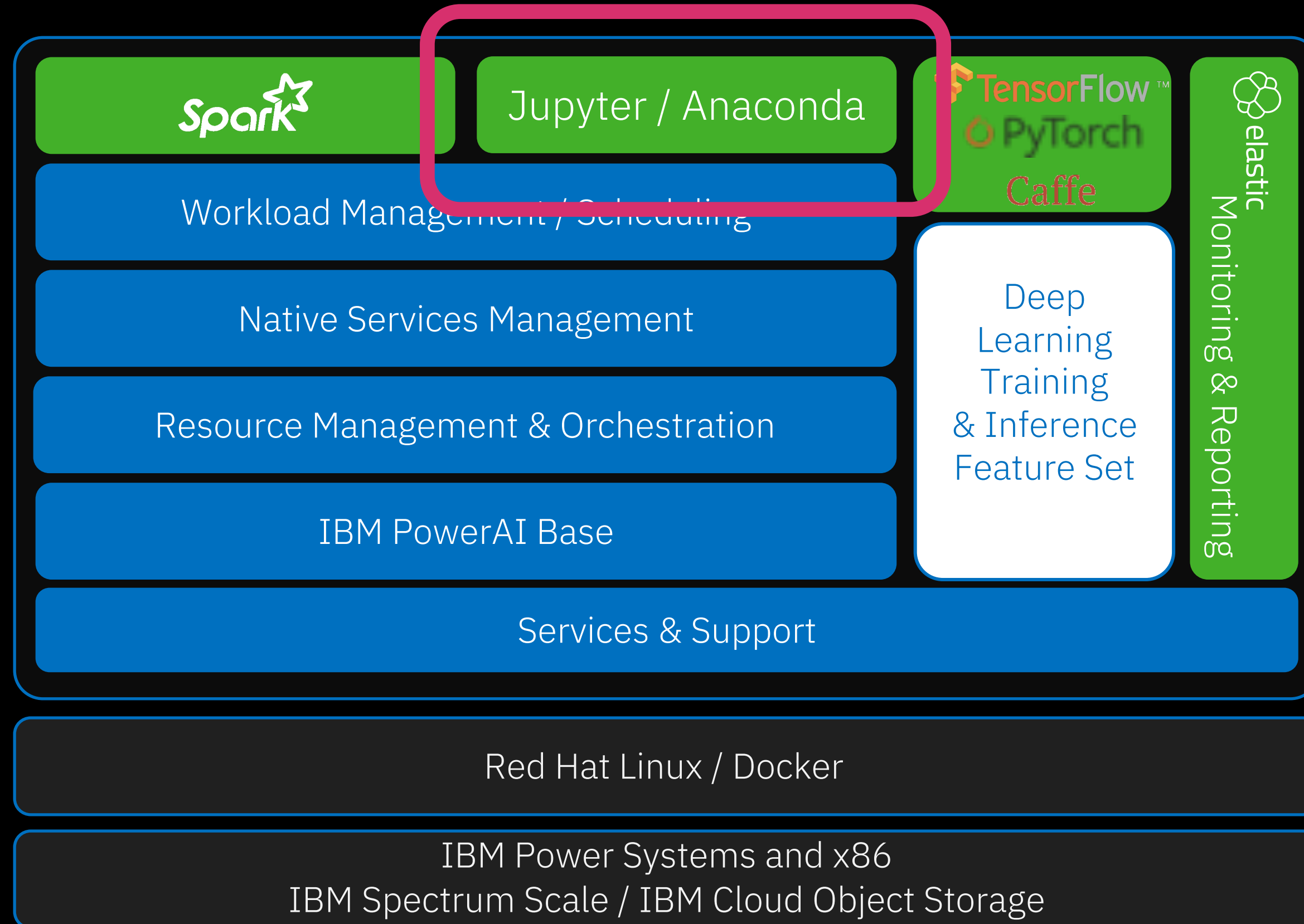
Elastic Distributed Training Example

- Single-node, multi-GPU training support without parameter server .

- Multi-node, multi-GPU training support with parameter server.

- Both synchronous and asynchronous distributed training algorithms support for multi-node, including synchronous gradient data control algorithm, and asynchronous gradient data control algorithm.

- NVIDIA Collective Communications Library (NCCL) support for broadcasting, reducing gradient and weight data across multiple GPUs.

- Review modifications for model for EDT

More Info:

Edit model for elastic distributed training
https://www.ibm.com/support/knowledgecenter/SSFHA8_1.2.0/us/configure-pytorch-model-elastic-distributed-training.html

Jupyter / Anaconda

Spark

Jupyter / Anaconda

TensorFlow™
PyTorch
Caffe

elastic
Monitoring & Reporting

Workload Management / Scheduling

Native Services Management

Resource Management & Orchestration

IBM PowerAI Base

Deep Learning Training & Inference Feature Set

Services & Support

Red Hat Linux / Docker

IBM Power Systems and x86
IBM Spectrum Scale / IBM Cloud Object Storage

Jupyter / Anaconda Overview

- Distributed anaconda distribution, conda environment, and conda package management
- Option to link to Jupyter notebook

  - Jupyter notebook linked to underlying conda environment

  - Conda environment managed per group and ability to create multiple

  - Collaboration option

  - Easy interface into cluster (development, data management / cleansing, etc.)

More Info:

Notebooks (snapml, xgboost, keras, and more)
https://github.com/IBM/wmla-assets

Tutorials
https://developer.ibm.com/tutorials/category/watson-ml-accelerator/

Open a notebook and create a note
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/tutorial/t_notebook.html

Work with anaconda through notebook

- Workload > Spark > My Application and Notebooks > Open Notebook
  - If you don't see this, maybe your users needs a notebook assigned to it or collaboration enabled

More Info:

Notebooks (snapml, xgboost, keras, and more)
https://github.com/IBM/wmla-assets

Tutorials
https://developer.ibm.com/tutorials/category/watson-ml-accelerator/

Open a notebook and create a note
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/tutorial/t_notebook.html

Work with anaconda through CLI

- Activate the Anaconda Distribution instance

```
source
/nfs/public/wmla/anaconda/[group]/201903/anaco
nda/etc/profile.d/conda.sh
```

- Activate the conda environment

```
conda activate dlipy3
```

More Info:

Notebooks (snapml, xgboost, keras, and more)
https://github.com/IBM/wmla-assets

Tutorials
https://developer.ibm.com/tutorials/category/watson-ml-accelerator/

Open a notebook and create a note
https://www.ibm.com/support/knowledgecenter/en/SSZU2E_2.3.0/tutorial/t_notebook.html

Open Discussion / Questions
Thank you!!

Eric Fiala
Cognitive Solution Architect
—
Eric.J.Fiala@ibm.com
ibm.com

Continue the discussion on Slack
ibm-systems-cern.slack.com
#cern-openlab