DELL Technologies

Whitepaper

Reference Architecture for End-to-End Data Science at Scale

On-premises data science platform for machine learning and deep learning using Iguazio

Abstract

This whitepaper outlines the technical considerations for deploying an on-premises enterprise data science reference architecture jointly developed by Dell Technologies, Intel® and Iguazio® optimized for real world performance and scalability. The Iguazio Data Science Platform on Dell EMC infrastructure with the latest Intel Xeon® Scalable processors and NVMe storage enables organizations to implement optimized machine learning (ML) and deep learning (DL) projects faster and manage and scale them more easily.

February 2020

Contents

Executive summary

Organizations in many industries are recognizing the value of advanced computing models, such as artificial intelligence (AI), to generate competitive advantage from vast amounts of data. AI, supported by machine learning (ML) and deep learning (DL) applications. These models can cut costs and increase efficiency by reducing the need for human intervention, and/or by assisting in decision making for a variety of use cases.

However, many companies struggle with building the high performance computing (HPC) systems needed to support AI initiatives. Such systems can be complicated to implement and run, requiring skills that are outside the core competency of some enterprise IT departments.

To overcome these challenges and help make the advantages of AI more accessible, Dell Technologies has collaborated with Intel and Iguazio to develop an Iguazio-validated reference architecture based on the Iguazio Data Science Platform, Intel-optimized AI libraries and frameworks, and Dell EMC infrastructure.

The reference architecture enables data science teams to take advantage of the Iguazio Data Science Platform more quickly and easily, leveraging validated configurations of industry-leading Dell EMC servers with 2nd Generation Intel Xeon Scalable processors, Dell EMC networking and data storage.

Dell Technologies, Intel and Iguazio collaborate to develop validated reference architectures based on the Iguazio Data Science Platform with Intel-optimized AI libraries and frameworks.

Solution overview

Dell Technologies collaborates with Intel and Iguazio to create reference architectures specifically for Iguazio software, to enhance performance for AI and ML workloads that are critical for advancing business objectives. For added flexibility, the design for Iguazio uses a flexible building block approach to system design, where individual building blocks can be combined to build a system that's optimized specifically for your unique workloads and use cases.

The optimized solution blocks are shown in Figure 1. They include density-optimized Dell EMC PowerEdge C6420 servers for compute nodes, Dell EMC PowerEdge R740xd two-socket servers with Intel Xeon processors and Intel NVMe solid-state drives (SSDs) for data nodes, and high throughput networking infrastructure from to enable efficient execution of real-time and streaming AI uses cases.

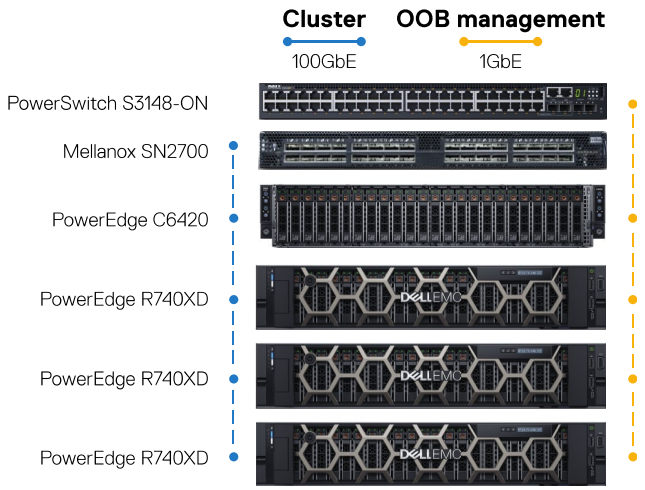
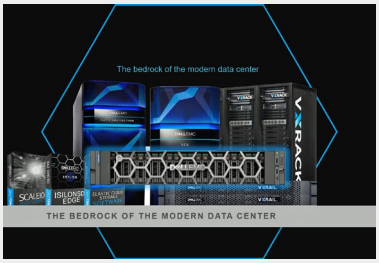


Figure 1. *Dell EMC reference architecture for Iguazio Data Science Platform building blocks*

Innovative designs to transform IT

As the foundation for a complete, adaptive IT solution, Dell EMC PowerEdge servers deliver dramatic performance and management advantages that more effectively and efficiently power the business applications you need to succeed.



Intel Xeon processors for AI

With recent advancements in hardware-based AI acceleration, software optimizations to AI frameworks, and specialized AI libraries, Dell EMC PowerEdge servers with Intel Xeon Scalable processors provide outstanding performance and scalability on the CPU platform people know and trust.

Dell EMC PowerEdge servers

Dell EMC PowerEdge servers with Intel Xeon Scalable processors, offer flexible choices to optimize performance for a variety of application types. One-socket servers provide balanced performance and storage capacity for future growth. Two-socket servers provide an optimum balance of compute and memory for most enterprise workloads. Four-socket servers provide the highest performance and scalability for advanced computing workloads. Built for scale-out workloads like HPC, AI and data analytics, Dell EMC PowerEdge C Series servers deliver the latest high-speed memory, fast NVMe storage and workload-based BIOS tuning. You can scale efficiently and predictably with flexible configurations and advanced connectivity options. Intelligent automation empowers your team, freeing them from routine maintenance.

2nd Generation Intel Xeon Scalable Processors

The 2nd generation Intel Xeon Scalable processors are optimized for demanding data center workloads. This processor family features higher frequencies than previous-generation Intel Xeon Scalable processors, along with architecture improvements.

The 2nd Generation Intel Xeon Scalable processors take AI performance to the next level with Intel Deep Learning (DL) Boost, which extends the Intel Advanced Vector Extensions 512 (Intel AVX-512) instruction set with Vector Neural Network Instructions (VNNI). Intel DL Boost significantly accelerates inference performance for DL workloads optimized to use VNNI—sometimes by as much as 30X compared to a previous-generation Intel Xeon Scalable processor.[1]

Intel Data Analytics Acceleration Library

Intel Data Analytics Acceleration Library (Intel DAAL) is an easy-to-use library that helps applications deliver predictions more quickly and analyze large data sets without increasing compute resources. It optimizes data ingestion and algorithmic compute together for high performance, and it supports offline, streaming and distributed usage models to meet a range of application needs.

Intel DAAL can help with these stages of analytics:

- Pre-processing (decompression, filtering and normalization)
- Transformation (aggregation and dimension reduction)
- Analysis (summary statistics and clustering)
- Modeling (training, parameter estimation and simulation)
- Validation (hypothesis testing and model error detection)
- Decision making (forecasting and decision trees)

New features include high-performance logistic regression, extended functionality and user-defined data modification procedures.

daal4py is a simplified Python® API to Intel DAAL that allows for fast usage of the framework. It provides highly configurable ML kernels, some of which support streaming input data and/or can be easily and efficiently scaled out to clusters of workstations.

[1] Intel brief, Second Generation Intel Xeon Scalable Processors, accessed January 2020.

Iguazio Data Science Platform software

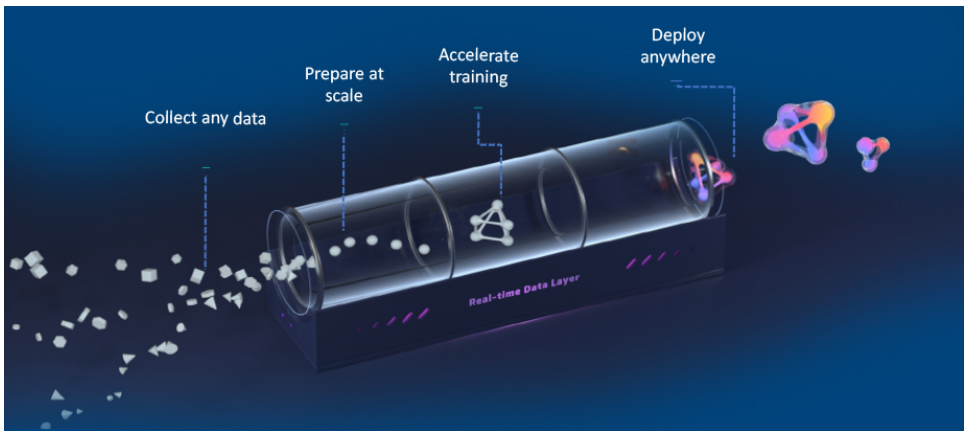
The Iguazio Data Science Platform software automates and accelerates ML workflows, enabling data scientists to develop, deploy and manage real-time AI applications at scale. Iguazio speeds and simplifies deployment of AI and ML applications by building in essential frameworks, such as Kubeflow, Apache® Spark® and TensorFlow™, along with well-known orchestration tools like Docker® and Kubernetes®. The Iguazio software platform enables simultaneous access through multiple industry-standard APIs for streams, tables, objects and files that are stored and normalized once, so you can launch new projects quickly and then consume, share and analyze data faster.

Iguazio enables real-time processing of streaming data for rapid time-to-insight. By unifying the data pipeline, Iguazio reduces the latency and complexity inherent in many advanced computing workloads, effectively bridging the gap between development and operations.

It also provides friendly pipeline orchestration tools, serverless functions and services for automation, and an extremely fast multi-model data layer, all packaged in a managed and open platform. Plus, it delivers fine-grained security using multi-layered network, identity, metadata or content-based policies.

iguazio

Bring your AI applications to life with Iguazio's Automated Data Science Platform software on Dell EMC PowerEdge Servers



Source: iguazio.com

Iguazio Data Science Platform software on Dell EMC infrastructure provides the following capabilities:

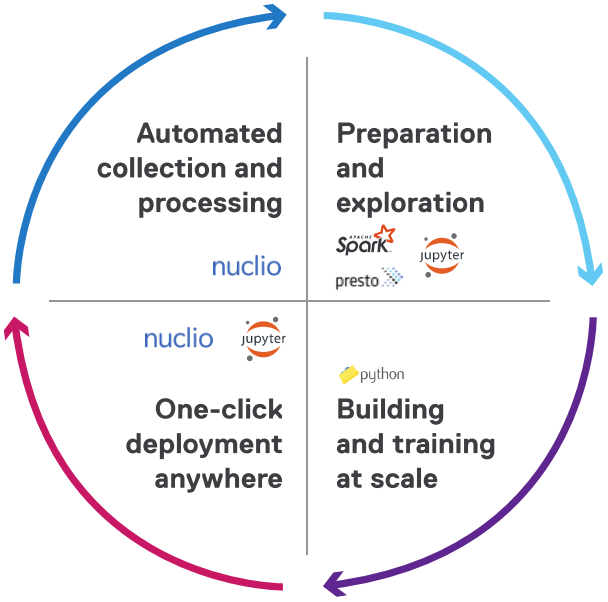
- **Collect and enrich data from any source**. Ingest multi-model data at scale in real time, including event-driven streaming, time series, NoSQL, Microsoft® SQL Server® and other files.
- **Prepare online and offline data at scale.** Explore and manipulate online and offline data using a real-time data layer and your preferred data science and analytics frameworks.
- **Accelerate and automate model training.** Continuously train models in a production-like environment, dynamically scaling GPUs and managed ML frameworks.
- **Deploy in seconds.** Deploy models and APIs from Jupyter® Notebook or IDE to production in just a few clicks and continuously monitor model performance.

Simplify and accelerate ML pipelines by using the Iguazio software platform to:

- **Manage end-to-end workflows.** Manage and automate the entire workflow, streamlining the process of data preparation, training, validation and deployment to operationalize models.

- **Track experiments.** Automatically track code, metadata, inputs and outputs of executions with MLRun and easily reproduce results.
- **Build an automated pipeline.** Developing with Iguazio engines enables building, training, optimizing and deploying models in a production-ready environment, while automating DevOps tasks.
- **Run feature engineering in real-time.** The platform provides real-time feature engineering based on fast data layer and other functions. Users can leverage the software platform capabilities of key value and time series with built-in aggregation functions to calculate and analyze data in real time.

The different open-source tools used for the stages of an ML pipeline work within the platform as follows.



Automated data collection and processing	Model preparation and feature exploration
Nuclio helps users collect data from various sources and types. Event-driven capabilities make it ideal for processing streaming data in real time. Users can leverage other frameworks, such as Spark, for data collection.	Users can pick and choose from popular tools for exploring and preparing data. It's common for data scientists to use Jupyter Notebook, which is already integrated with tools and libraries such as Pandas Spark, Presto and more. Storing and accessing data is done using different formats, such as NoSQL, time series, stream data and files. Multiple APIs are used to access and manipulate the data from diverse sources.
Building and training at scale	**One-click model deployment**
Data science teams need access to diverse software environments and scalable compute capability to build models for enterprise projects. A Python environment of Iguazio with built-in ML libraries like Scikit Learn, NumPy, PyTorch®, and TensorFlow over Kubernetes, helps users build and train models smoothly. Users can also leverage built-in distributed frameworks such as Dask and Horovod to run Python at scale.	Model deployment is typically handled by different teams using another set of tools, leading to projects being delayed. The Iguazio platform provides a smooth way to deploy code into a function that can easily run in a production pipeline without any DevOps, significantly reducing the level of effort required to deploy code in production.

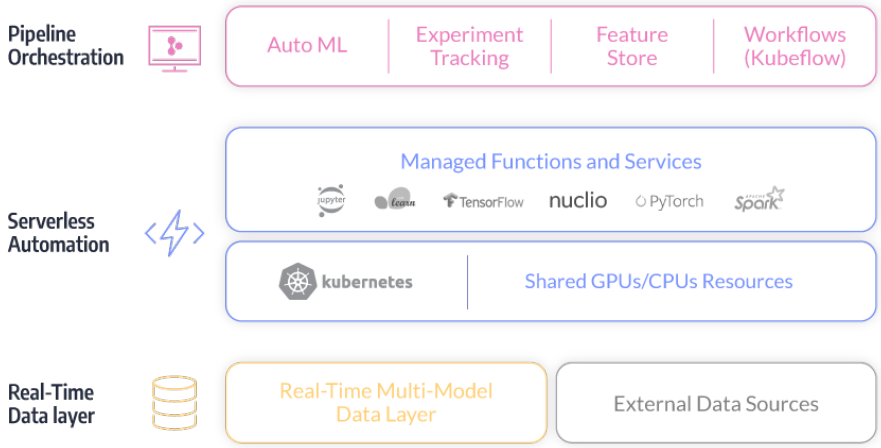
ML pipeline delivered in the cloud, at the edge or on premises

This unique, unified data model eliminates the need for multiple data stores, constant synchronization, complex pipelines and painful extract-transform-load (ETL) processes.

Iguazio Data Science Platform on Dell EMC infrastructure

The Iguazio Data Science Platform software on Dell EMC infrastructure consists of three main components, all packaged and managed in an open platform.

- **Real-time data layer:** Provides fast, secure and shared access to real-time and historical data running as in-memory databases on Flash memory, enabling low costs and high density.
- **Serverless automation:** Automates DevOps to save time and resources typically spent on data collection, packaging, scaling, tuning and instrumentation.
- **Pipeline orchestration:** Provides end to-end workflow management via a full-stack, user-friendly environment featuring fully integrated workflow management, experiment tracking and AutoML tools.



Source: Iguazio, Ltd.

***Figure 2**: Architecture and key components of the Iguazio Data Science Platform software*

Real-time data layer

Data is the fuel that powers ML and DL techniques. In addition to connecting to external data sources used for data engineering and curation to build datasets, the platform has a built-in, multi-model data layer, also called a data store or database.

The built-in data layer is used for storing and analyzing various types of data structures — such as NoSQL tables, time-series databases, data streams, binary objects and files. The data can be accessed through multiple industry-standard and industry-compatible programming interfaces. In fact, data ingest can be through one interface and consumed through another interface depending on preferences and needs. This unique, unified data model eliminates the need for multiple data stores, constant synchronization, complex pipelines and painful extract-transform-load (ETL) processes.

Leverage the key value and time-series database for storing and computing online features. Iguazio provides built-in primitives that enable real-time feature engineering at scale. Store and analyze columnar data by leveraging data formats such as Parquet or Avro running on the data fabric, mainly for offline training.

As part of the data fabric, Iguazio has a time-series database. The time-series database includes a rich set of features for efficiently analyzing and storing time-series data. The software platform uses the Iguazio V3IO TSDB open-source library, which exposes a high-performance API for working with TSDBs — including creating and deleting TSDB instances and ingesting and consuming TSDB data. Work with Prometheus for ingesting and fetching data from the time-series database.

Serverless automation

Serverless automation accelerates and facilitates the development process by eliminating many server orchestration tasks and allowing developers to concentrate on development logic. Developers can embed flexible computation within their data flow without worrying about server infrastructure provisioning and/or DevOps considerations.

Serverless technologies allow developers to write code and specifications, which are then automagically translated to auto-scaling production workloads. Until recently, these were limited to stateless and event driver workloads, but now with the new open-source technologies like MLRun, Nuclio and Kubeflow, serverless functions can take on the larger challenges of real-time, extreme scale data analytics and machine learning.

Managed functions and services

Iguazio Data Science Platform software comes with essential and useful open-source tools and libraries that facilitate the implementation of a full data science workflow, from data collection to production. Both built-in and integrated tools are exposed as application services that are managed by the platform using Kubernetes.

Each application is packaged as a logical unit within a Docker container and is fully orchestrated by Kubernetes, which automates the deployment, scaling, and management of each containerized application. This provides users with the ability and flexibility **to run any application anywhere**, as part of their operational pipeline.

The application services can be viewed and managed from the dashboard using a self-service model. This approach enables users to quickly get started with their development and focus on the business logic without having to spend precious time deploying, configuring, and managing multiple tools and services. In addition, users can independently install additional software — such as real-time data analytics and visualization tools — and run them on top of the software platform services.

Get started quickly with development and focus on the business logic without having to spend precious time deploying, configuring, and managing multiple tools and services.

Some of the key services include:

- Jupyter lab with a full Python environment
- Apache Spark
- Presto (distributed SQL engine)
- Nuclio
- MLRun
- Grafana (for visualization)
- Frames (high speed library)
- REST API service
- Kubeflow pipeline

The software platform has built-in monitoring and logging services that capture telemetry and logs across services.

Using Nuclio with Jupyter Notebook

Nuclio can be easily integrated with Jupyter Notebook, enabling developers to write their entire code in Jupyter Notebook and use a single command to deploy it as a serverless function that runs in the serving layer. For examples, see the Jupyter Notebooks tutorial in GitHub. For more information about Nuclio, see the Serverless Functions (Nuclio) introduction.

Serverless functions using Nuclio

To provide users of the Iguazio Data Science Platform software with a serverless solution, Iguazio developed Nuclio — a stand-alone, open-source, self-service application-development environment. Nuclio Enterprise edition is integrated into the software platform. It allows developers to build and run auto-scaling applications in their preferred programming language, without worrying about managing servers.

Nuclio is currently one of the fastest serverless frameworks in the market. It allows runs functions over CPUs or accelerators, supports a large variety of triggers, can be deployed in the cloud or on-premises, and provides many other significant advantages. Nuclio provides users with a complete cloud experience of data services, ML and AI, and serverless functionality — all delivered in a single integrated and self-managed offering at the edge, on-premises, or in a hosted cloud.

Within the data science flow, you can use Nuclio for functions such as:

- **Data collection and preparation.** Collect, ingest and consume data on an ongoing basis. Nuclio offers built-in function templates for collecting data from common sources, such as Apache Kafka® streams or databases, including examples of data enrichment and data pipeline processing. Users can add their logic for data transformation and manipulation.
- **Running models.** Run ML models in the serving layer supporting high throughput on-demand and elastic resource allocation.

Kubeflow Pipelines is an open source framework for building and deploying portable, scalable ML workflows based on Docker containers. The platform has a pre deployed tenant wide Kubeflow Pipelines service that can be used to create and run ML pipeline experiments. For detailed information, see the Kubeflow Pipelines documentation.

Pipeline orchestration

Iguazio leverages MLRun, an open-source pipeline orchestration framework. This is an easy-to-use mechanism for data scientists and developers to describe and run ML-related tasks in various scalable runtime environments. Data scientists and developers can also run ML pipelines while automatically tracking code, metadata, inputs, and outputs of executions. MLRun integrates and uses Nuclio serverless project and KubeFlow components. It tracks various elements, stores them in a database and presents running jobs, as well as historical jobs in a single report.

MLRun enables developers to run the same code either locally on a PC for a test or on a large-scale Kubernetes cluster with minimal changes. It also enables tracking experiments with their parameters, inputs, outputs and labels. The key elements of MLRun are:

- **Function.** A software package with one or more methods and runtime specific attributes. Function can run one or many tasks, they can be created from templates and stored in a versioned database.
- **Task.** Define the desired parameters, inputs and outputs of a job/task. The task can be created from a template and run over different runtimes or functions.
- **Run.** The result of running a task on a function, it has all the attributes of a task plus the execution status and results.

MLRun enables automated and scalable pipeline orchestration via:

- **AutoML.** Run multiple experiments in parallel, each using a different combination of algorithm functions and parameter sets (hyper-parameters) to automatically select the best result.

- **Experiment tracking.** Describe and track code, metadata, inputs and outputs of ML-related tasks (executions) and reuse results with a generic and easy-to-use mechanism.
- **Online and offline feature store.** Maintain the same set of features in the training and inferencing (real-time) stages with MLRun's unified feature store.

  Workflows (Kubeflow). Natively integrate with Kubeflow Pipelines to compose, deploy and manage end-to-end ML workflows with UI and a set of services.

Reference architecture and implementation

The reference architecture described in this document is an engineering-validated configuration of the Iguazio Data Science Platform in a high availability (HA) cluster built with Dell EMC PowerEdge servers with Intel processors and NVMe storage.

It was designed as a shared-nothing distributed architecture with no single point of failure. Every cluster has at least three data nodes and supports data replication, ensuring HA.

Configuration overview

Server	Dell EMC PowerEdge C6420	3x Dell EMC PowerEdge R740xd
Processor	Intel Xeon Scalable Gold 6248 @ 2.50GHz	
DRAM	192GB	576GB
Storage	Boot: 2x 480GB SSD	
	N/A	Dell Express Flash NVMe P4610 1.6/3.2TB SFF (6 to 24)
Network	Mellanox ConnectX®-4 100GbE QSFP adapter	PowerSwitch S3148-ON Ethernet 10G 4P X710 SFP+ rNDC with 4 ports
	Intel Gigabit I350-t LOM (1 each)	Mellanox ConnectX-4 100GbE QSFP adapter
Operating system	Red Hat® CentOS® or Red Hat Enterprise Linux® 7.6	
Software	Iguazio Data Science Platform, including Jupyter, Nuclio, MLRun, Grafana, Prometheus, Pipelines, Kubernetes, data containers and more.	

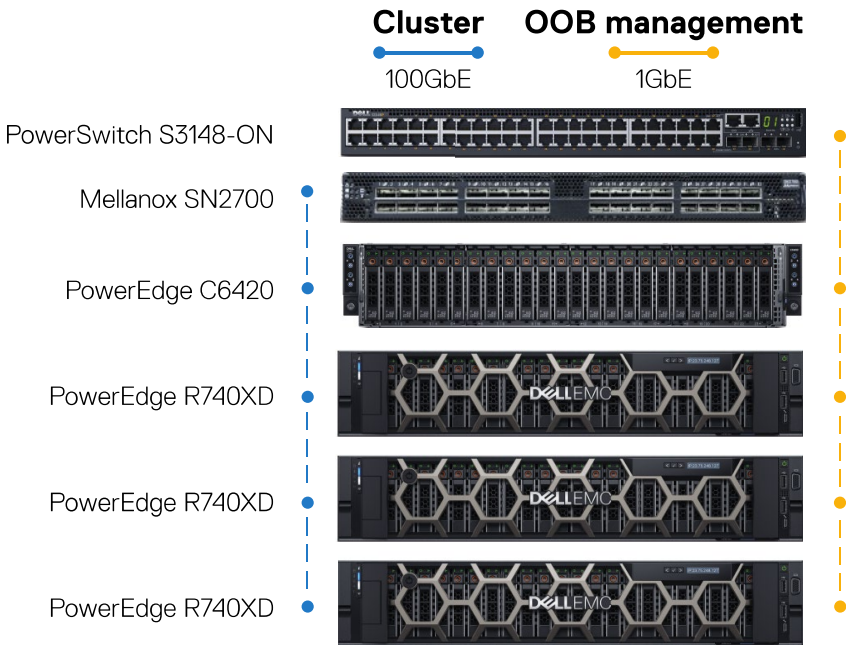


***Figure 3**: Hardware configuration*

The cluster design and components were chosen to maximize CPU utilization and leverage the benefits of non-volatile memory, remote direct memory access (RDMA) and Intel P4610 NVMe SSDs with dense storage. Dell EMC PowerEdge C6420 and R740xd servers configured with second generation Intel Xeon Scalable processors and Intel P4610 NVMe drives support the platform's performance requirements.

The Dell EMC PowerEdge C6420 is a density-optimized server that scales efficiently and predictably while drastically reducing complexity. It offers up to four independent hot-swappable two-socket servers in a very dense 2U package. Plenty of compute, memory, storage, connectivity and chassis options make it the ideal compute node for Iguazio workloads.

The Dell EMC PowerEdge R740xd is a two-socket, 2U rack server designed to run complex workloads using highly scalable memory, I/O capacity and network options. It offers extraordinary storage capacity options, making it well suited for data-intensive applications that require greater storage, while not sacrificing I/O performance.

Benefits of Iguazio delivered on Dell EMC infrastructure

Dell EMC has developed reference architectures with Iguazio to enable organizations to accelerate their AI transformation. Dell EMC Consulting also provides data analytics and AI services, from strategy through to implementation and ongoing optimization. It also helps to bridge the people, process and technology needed to achieve desired business outcomes with speed and scale. This includes implementing and operationalizing AI technologies and helping customers accelerate their data engineering capabilities.

Two main challenges of every organization when adopting machine learning in production is:

1. Significantly reduce time to market
2. Minimize the amount of resources and skill level needed to complete the project

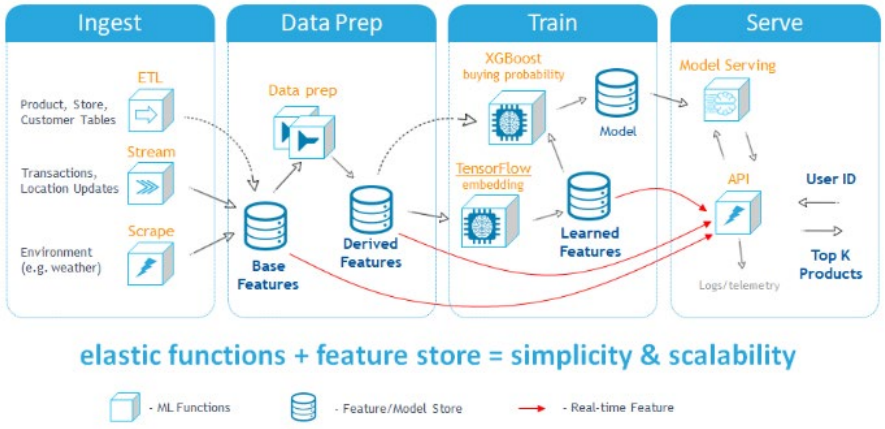
Learn more

- PowerEdge Reference Architectures
- Delltechnologies.com/ai
- Delltechnologies.com/servers
- Iguazio Data Science Platform
- Intel Deep Learning Boost
- Intel Deep Learning Reference Stack
- Intel AI Builders

Large organizations stand-up large software teams to develop platforms and processes to automate the steps of packaging, scaling, tuning, instrumentation, and continuous delivery. These steps are fully automated in the Iguazio software platform using innovative technologies to help address the challenges faced by organizations.

Iguazio reduces ML pipelines complexity by adopting the concept of "serverless ML Functions." These allow you to write code and specifications which automagically translate to auto-scaling production workloads. Until recently, these were limited to stateless and event driver workloads, but now with the new open-source technologies (MLRun+Nuclio+KubeFlow), these functions can take on larger challenges of real-time, extreme scale data-analytics and machine learning.

ML functions can easily be chained to produce ML pipelines (using KubeFlow). They can generate data and features which will be used by subsequent stages. The following diagram demonstrates the pipeline used to create a real-time recommendation engine application using the Iguazio Data Science Platform software.



Source: Iguazio, Ltd.

Security and compliance are top of mind for any IT solution. Iguazio Data Science Platform software implements multiple mechanisms to secure access to resources and keep data safe. Security management is available from a single pane of glass. The configuration is done in one user-friendly platform dashboard, and applied across all platform interfaces. The result is a significantly simplified, yet robust, data-security that helps organizations meet compliance objectives.

The software platform allows you to define local users and import users from an external identity provider, authenticate user identities and control user access to resources, including the ability to define fine-grained data-access policies. To ensure proper security, the platform uses time-limited sessions and supports the HTTP Secure (HTTPS) protocol.

Accelerate AI transformation with Dell Technologies, Intel and Iguazio

The reference architecture for Iguazio on Dell EMC infrastructure is designed to accelerate AI transformation. It significantly reduces time to production and minimizes the amount of resources and skill level needed to complete AI projects. Once operational, the Iguazio software platform automates the steps of packaging, scaling, tuning, instrumentation and continuous delivery, dramatically easing the burden on IT teams.

Assistance when you need it

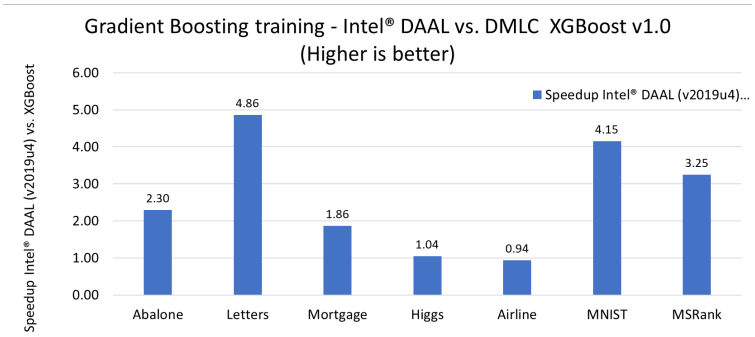
Dell Technologies Consulting provides data analytics and AI services, from strategy through to implementation and ongoing optimization. It helps to bridge the people, processes and technology needed to achieve desired business outcomes with speed and scale. This includes implementing and operationalizing AI technologies, and helping accelerate data engineering capabilities.

Appendix A

Deploy the Intel-optimized Software stack on Iguazio

Intel is continually developing optimized software stacks for HPC and AI workloads. Some of the critical software in AI space includes Intel-optimized Tensorflow, optimized Xgboost, and, most importantly, Intel Distribution for Python that packages Intel MKL library, optimized Numpy, scipy, scikit-learn, and Intel DAAL. Intel's optimized software stacks for AI are available in several distribution channels, including docker hub. As Iquazio's core process is based on docker containers, using Intel's optimizations simplify and streamline tasks and services.

The Intel DAAL library addresses stages of the data analytics pipeline: preprocessing, transformation, analysis, modeling, validation, and decision-making. It outperforms other solutions for developers and data scientists. The benchmark results below (tested conducted by Intel on 08/22/2019) compare performance of the XGBoost implementation in Intel DAAL to an XGBoost open source project. The y-axis shows a speedup factor of up to approximately five times speedup for four representative classification and regression test cases.



To use Intel Distribution for Python (IDP), deploy the use case with the IDP docker image on worker nodes, thereby accelerating serial and distributed computing. Intel Distribution for Python ships a highly-performant, easy-to-use data analytics library Intel DAAL, a Python Library to perform efficient classical Machine Learning tasks on single and multi-node environments at ease. One other unique feature in IDP is, Intel DAAL optimizes scikit-learn calls. Users need to set the "USE_DAAL4PY_SKLEARN" environment variable to "YES" to dispatch daal calls for sklearn. More details on daal-optimized sklearn can be found at https://intelpython.github.io/daal4py/sklearn.html

- On the Jupyter lab, add the below instructions

1. Use the Intel Python docker image

   1.1 Import Nuclio and config to use Intel Python core image.

```
import nuclio
%nuclio config spec.build.baseImage =
"intelpython/intelpython3_core:latest"
```

1.2 Install additional packages as needed

```
%%nuclio cmd
pip install --ignore-installed PyYAML
pip install mlrun==0.4.3
pip install matplotlib
pip install kfp
```

if you are using the core image, install additional packages such as daal and scikitlearn

```
conda install -y scikit-learn daal4py -c intel
```

1.3 Converted code from notebook will run on the base image after deployment

```
train_fn =
code_to_function(filename='higgs_xgb_srvless-intel-
daal-original-idp.ipynb', name='intel-higgs',
runtime='nuclio:mlrun')
```

(or)

1.4 You can also create a yaml file with the Intel Python image and the application code, and run it as a job

```
train_fn=code_to_function(handler='train_higgs',
                          kind='job')
train_fn.spec.build.base_image =
'intelpython/intelpython3_full:latest'
gen_fn.export('train_higgs.yaml')
```

2. Alternatively you can use a docker image created by Iquazio that packages mlrun, Intel's full Python Distribution, dask, xgboost and many more packages. This Docker image doesn't require additional Python package installs

```
import nuclio
%nuclio config spec.build.baseImage = "yjbds/mlrun-
daskboost"
```

Below is code snippet employing Intel DAAL and Iquazio on the widely-used dataset HIGGS. With this trivial change, one can gain all the necessary optimizations and accelerations through Intel DAAL on Iquazio. The succeeding steps of running tasks and deploying the use case remain unaltered.

```
import os
import daal4py as d4p
os.environ['USE_DAAL4PY_SKLEARN']= 'YES'
def higgs_train(

    context,
    dataset,
    model_name=MODEL_NAME, # model's file name with
extension
    target_path='',        # destination folder of model
    key='',                  # name of model in artifact
store
    maxTreeDepth=6,
    observationsPerTreeFraction=1,
    nClasses=2,
    steps=20,
    tree="hist",
    test_size=0.33,
    ftype='float',
    rng=7
):

    dataset =
pd.read_parquet(str(dataset),engine='pyarrow')
    dataset.dropna(axis=0, inplace=True)
    Y = dataset.pop('labels')
    test_size = test_size
    X_train, X_test, y_train, y_test =
train_test_split(dataset, Y,

test_size=test_size,

random_state=rng)
    y_train = y_train[:, np.newaxis]
    y_test = y_test[:, np.newaxis]

    # Get params from event
    param = {
        'fptype':                       ftype,
        'maxTreeDepth':                 maxTreeDepth,
        'minSplitLoss':                 0.1,
        'shrinkage':                    0.1,
        'observationsPerTreeFraction':
observationsPerTreeFraction,
        'lambda_':                      1,
        'maxBins':                      256,
        'featuresPerNode':              0,
        'minBinSize':                   5,
        'minObservationsInLeafNode':    1,
        'nClasses':                     nClasses}
```

```
    # Train model
    train_algo = d4p.gbt_classification_training(**param)
    train_result = train_algo.compute(X_train, y_train)
    #Predict
    predict_algo =
d4p.gbt_classification_prediction(param['nClasses'],
fptype=ftype)
    preds= predict_algo.compute(X_test,
train_result.model).prediction.ravel()
    best_preds = np.asarray([np.argmax(line) for line in
preds])

    # log results and artifacts
    context.log_result('accuracy',
float(accuracy_score(y_test, best_preds)))

    filepath = os.path.join(target_path, model_name)
    dump(train_result.model, open(filepath, 'wb'))
    context.log_artifact('model', target_path=filepath,
labels={'framework': 'GBT'})
```