DELL Technologies

Whitepaper

Reference Architecture for Data Science Platform Using Kubeflow

Blueprint for open-source machine learning platform on Kubernetes

Abstract

This paper assembles the experience of Canonical®, Dell, SUSE®, Intel® and Grid Dynamics® in designing, building and supporting machine learning (ML) and data science platforms over the years. It covers typical business problems, IT operations complexity and potential pitfalls that you will face while building your own machine learning or data science platform—and provides guidance and solutions to help you overcome and avoid many of those issues, saving time and effort.

February 2020

Grid Dynamics

© 2020 Dell Inc. or its subsidiaries.

Contents

© 2020 Dell Inc. or its subsidiaries.

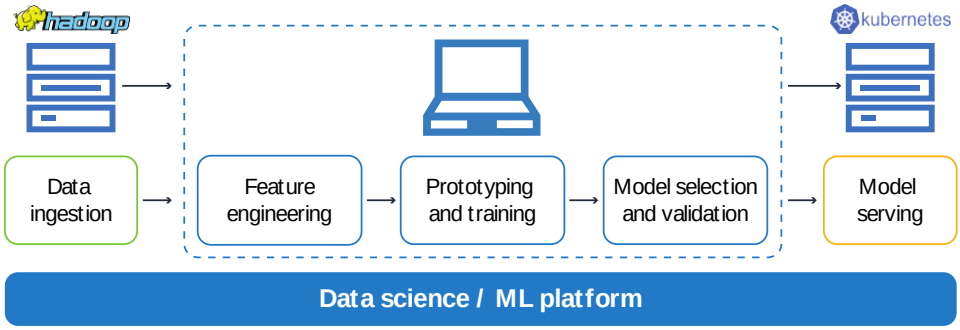
Complexity of building ML platform

Despite a rich set of ecosystem tools for data science and machine learning, there is still a big gap between the benefit it can provide and the real business needs. There are a limited number of services that can effectively cover such system qualities as availability, release management or even basic security. And that does not include IT operational aspects — integration with enterprise systems such as data warehouses, data lakes, user management systems, monitoring and auditing.

In the last few years we have witnessed a huge technical leap in data processing (Apache® Hadoop® ecosystem, Apache Kafka®, Apache Beam®, etc.) and compute (Kubernetes®, functions as a service, on-demand computation, etc.). However, when we take a look at the data science and machine learning domain, we need a solution that goes beyond the data science engineer's laptop and the "works on my machine" attitude.

Despite a recent plethora of new frameworks being introduced (TensorFlow™, Caffe2, PyTorch®) and the usage of accelerators and AI on CPUs, this ecosystem is lagging compared to the mature Hadoop ecosystem.

Therefore, when we look at a typical data science development lifecycle, we observe an inefficiency or bottleneck due to data science tasks being performed on non-performant infrastructure (such as the data scientist's laptop).



In the case of the biggest service providers, this phenomenon is well known and described. It is unfortunately common for systems that incorporate machine learning methods to end up with high-debt design patterns.

© 2020 Dell Inc. or its subsidiaries.

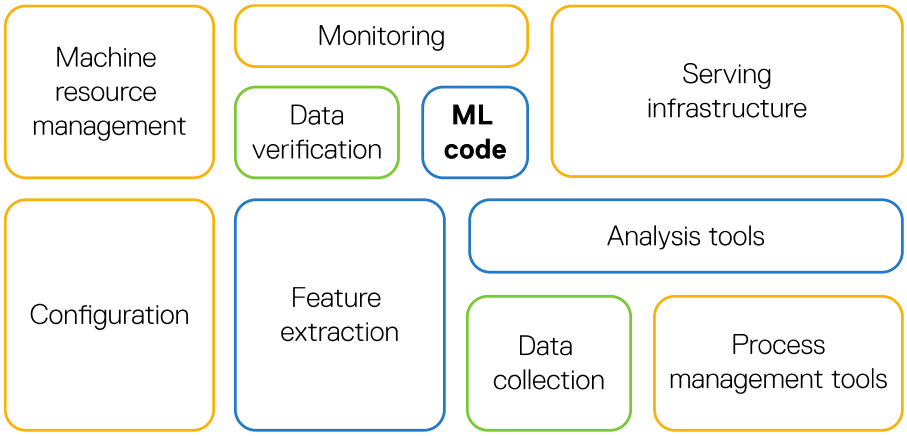


***Figure 1**: Only a small fraction of a real-world ML system is composed of the ML code, as shown by the small box in the middle. The required surrounding infrastructure is vast and complex.*[1]

Glue code, pipeline jungles, and dead experimental code paths in addition to configuration debt are some reasons why the market doesn't have a widely adapted product for machine learning and data science tasks beyond the engineer's laptop and Jupyter® Notebooks.

Motivation and business drivers

In addition to the issues described above, there are other motivations to building a machine learning platform as a standalone option to be easily integrated with existing on-premises data center infrastructure.

From Jupyter Notebook to production cluster

As previously mentioned, a lot of work related to data science happens on engineers' laptops using Jupyter Notebooks. This creates a custom environment difficult (or impossible) to move into production.

In the case of data science platform services from cloud service providers such as Amazon Web Services® (AWS), Microsoft® Azure® or Google® Cloud Platform™ (GCP), these are predominantly "outsourced" Jupyter Notebooks with custom integration to their own ecosystem (plugins). This allows them to deal with problems on a certain level, which is hard to replicate not only on-premises but also on another cloud provider. Being cloud-agnostic and avoiding vendor locks are very important and a feature in the proposed data science platform solution using Kubeflow.

From public cloud service to product

Commonly, it is not possible to mimic a public cloud solution because it relies on services unique to the service providers, such as GCP (Borg, Jupiter, Colossus). Similar services exist in AWS and Azure but with less impact. Those service providers lay down critical components, such as networking, storage subsystems, and file systems, and they can't be replicated at all. This happens when a "custom build" solution becomes a service without having an intervening production phase.

[1] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," Google, Inc., 2015.

© 2020 Dell Inc. or its subsidiaries.

Kubeflow, one of the most critical components of our platform, is a good example of such a service. Despite working well in a GCP environment, we faced numerous issues adopting it to typical data center needs. Some examples of products that can be successfully adopted in a private data center are the Hadoop ecosystem (Apache HBase®, Hadoop Distributed File System [HDFS]) and Kubernetes.

Easily integrating a machine learning platform into a data center ecosystem has many benefits. It naturally fills the gap between the data processing platform and the serving environment, and it can be the only solution the client would ever need.

Data locality and network

Though hybrid cloud is the preferable approach (and a profitable one) for common workloads, there are cases when it does not work as well, mostly related to the network. The network is often the slowest and least reliable component of any complex system. And in data processing it is critical.

This original paper on HDFS states, “An important characteristic of Hadoop is the partitioning of data and computation across many (thousands) of hosts, and executing application computations in parallel close to their data.” You bring computation to your data, so you don’t need to transfer huge amounts of data across network.

A hybrid cloud network can be unreliable and expensive. If data is stored and processed in different places, there can be throughput problems and big expenses related to data transfer. As a result, keeping data close to the processing systems—working with huge data sets without transferring them across many miles of Fibre Channel—is vital.

Many clients are keeping their data on-premises for security reasons, not wanting to store it on “someone else’s computer.” They question using remote services for processing in terms of both security and efficiency.

Here is where an in-house machine learning platform shines. The integration with common data sources and common enterprise services make it possible to extend data plane functionality and build an in-house analytics platform without external dependencies.

Changing landscape

Despite trends related to NoSQL- and Hadoop-based solutions a trend is emerging toward SQL-compatible solutions. Distributed systems lack many features of data warehouses.

In a world where schema-less data lakes dominate and most data is in tables, Python® is becoming the new SAS language and Jupyter Notebook the modern business intelligence tool. A machine learning platform can not only be an extension for the data lake but can also become the user interface for the next generation of data warehouse.

© 2020 Dell Inc. or its subsidiaries.

Blueprint

Any system that hopes to resolve the problem put forth in "Hidden Technical Debt in Machine Learning Systems" should satisfy a list of requirements. A machine learning platform should be able to:

- Cover the following operational needs:
  - Resource management — allocate the CPU/memory/disk needed to execute, provide access to accelerators, GPUs, etc.
  - Security management — user management, encryption in motion and at rest, BCDR capability
  - Release management — deployments, rollout/rollback capabilities and resilience for SLA-driven workloads
  - Application management — SRE drivers
- Easily integrate with critical enterprise services, such as AD/LDAP, monitoring and log management
- Integrate with typical data sources, such as SQL databases, data warehouses and data lakes
- Run widely adopted data processing frameworks and solutions, such as Apache Spark®, Apache Flink®, Apache Kafka®
- Run data analytics and machine learning frameworks and services, such as Jupyter servers, XGBoost, scikit-learn, TensorFlow, PyTorch and their Intel-optimized versions
- Allow integration of Intel-optimized libraries for faster performance of both training and inference on Intel Xeon® Scalable processors.
- Provide an API, SDK and web user interface to consumers, such as developers and operators

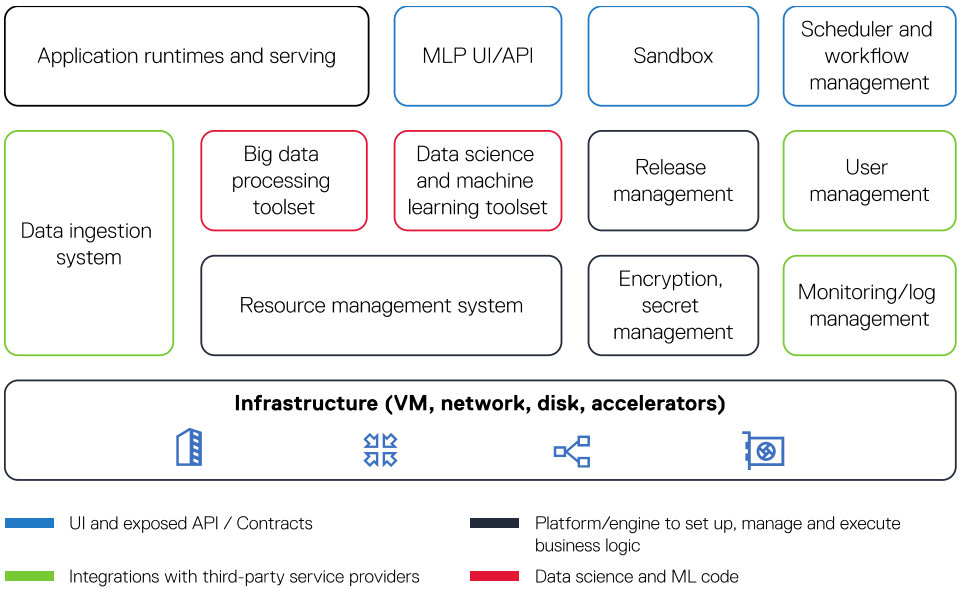


Figure 2: *Machine learning platform coverage*

© 2020 Dell Inc. or its subsidiaries.

After multiple iterations and development cycles, we created a blueprint that proves to be effective and efficient across many projects. The blueprint consists of different components that are described next.

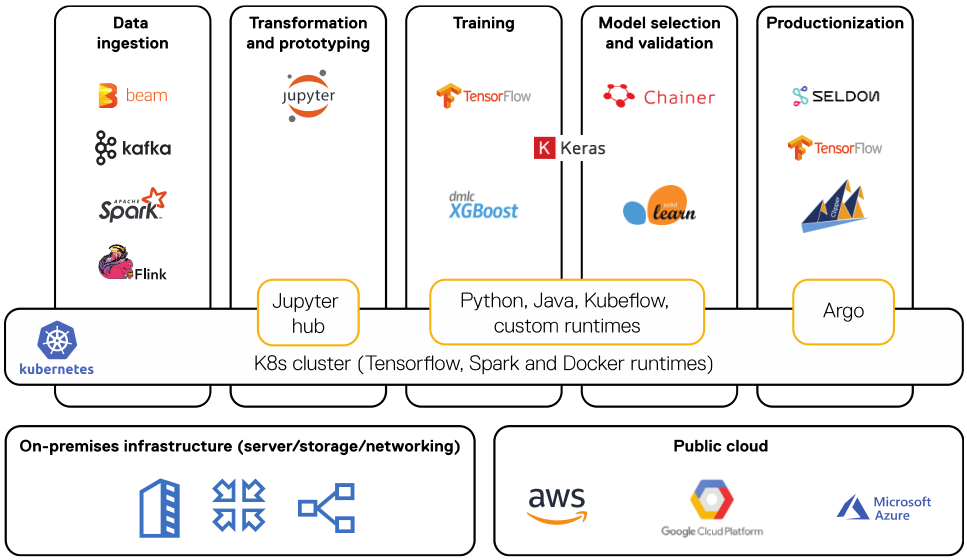


Figure 3: *Machine learning platform blueprint*

Hardware infrastructure

We start with a solid Dell hardware base, including servers, network equipment and — most important — experts across Dell, Intel and Grid Dynamics who help design, build and deliver a data science platform leveraging open-source AI and ML application stacks. This Reference Architecture provides a foundation for those looking to:

- Migrate an existing data platform based on open-source technologies
- Move from a legacy platform based on proprietary products
- Implement a new cloud-native platform
- Implement a specific capability on a cloud-native stack

Reference architecture components

High availability Kubernetes	Application nodes	Ceph storage	Networking	Software
3x PowerEdge R640	4x PowerEdge R740xd	3x PowerEdge R740xd2	1x PowerSwitch S3048-ON 1GbE 2x PowerSwitch S5248-ON 10/25GbE	Kubeflow on SUSE CaaS Platform Kubeflow on Canonical Charmed Kubernetes Cluster

© 2020 Dell Inc. or its subsidiaries.

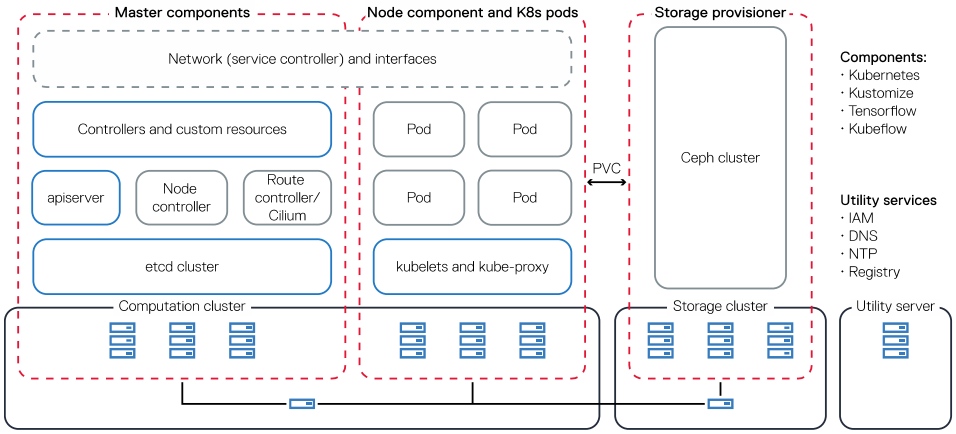


Figure 4: *High level diagram of solution*

Kubernetes as the control center

Kubernetes is the main control plane and central API for practically everything built further up the stack. There are a few reasons for this decision:

- Kubernetes architecture is heavily inspired by the concept of declarative application management, which gives us the ability to manage different common off-the-shelf (COTS) components.
- Though Kubernetes is one of many solutions for orchestration, its biggest benefit is being a universal API for resource management.
- Another important aspect is its ability to extend its API using custom resources and operators without increasing complexity or creating tech debt.
- The final reason is that Kubernetes is already widely adapted across the industry, so no special skills are required to operate it.

Thus, we use Kubernetes and collaborate with Canonical and SUSE to deliver an optimized, validated and automated Kubernetes solution. Because of the complexity associated with a machine learning platform, clients will benefit from first line support for the entire solution stack, including automated installation of low-level infrastructure components.

To speed deployment and time to solution of an on-premises data science platform, the reference architectures from Canonical and SUSE are a perfect starting point to build your own solution:

- Canonical/Dell Reference Architecture
- SUSE/Dell Reference Architecture

Data science and machine learning frameworks

Two things we need from our cluster operating system are to manage the hardware infrastructure and provide the interface to applications — which consists of a plethora of pre-processing, analytical and machine learning frameworks and services. Kubernetes custom resources and operators allow this conveniently and natively.

The most notable and demanded services are:

- Apache Spark and Apache Flink for big data pre-processing
- TensorFlow Extended, PyTorch and XGBoost for machine learning and data analytics
- Scikit-learn, Keras and Dask

© 2020 Dell Inc. or its subsidiaries.

There are a few dozen other purposely built libraries and frameworks that can be easily added to the system using Docker deployment options — Docker® images.

Intel offers libraries and frameworks with software optimizations to deliver faster AI training and inference with their world-class Intel Xeon Scalable processors. These include TensorFlow, MxNet, PaddlePaddle, Caffe and PyTorch.

Integrating these into enterprise a data science and machine learning solution enables users to quickly deploy a reliable, powerful solution with better performance. The Intel distribution for Python for example accelerates AI-related Python libraries such as NumPy, SciPy, and scikit-learn with integrated Intel Performance Libraries such as Intel MKL to deliver faster AI training and inference.

Pipeline, schedulers and Kubeflow SDK

Users of a data analytics system are familiar with patterns such as ETL and concepts such as pipelines and schedulers. Implementing them reliably and efficiently is a challenge. Apache Airflow®, an open-source solution, covers those issues but is not well suited for Kubernetes. Additionally, its complexity may require resources that compete with the whole machine learning platform.

The Kubeflow project can deal with this problem without adding huge overhead on top of an existing Kubernetes cluster. It uses custom resources and operators to enrich whole ecosystems and does not “reinvent the wheel” in terms of operations to support machine learning. Additionally, it provides the ability to build and run pipelines using UI, Python SDK or Argo Workflows.

UI and Jupyter Hub

A complex machine learning platform needs a simple interface. For this, the Jupyter Project is ideal. The machine learning platform serves as a resource provider for Jupyter Hub and Jupyter Server, which the user can spin off on-demand using any default or custom Docker image.

Serving

The machine learning platform must also be able to deploy the results of analytics, making them quickly available to consumers. If the pipeline is continuous integration for data science, the ability to deploy results can be continuous deployment. This is accomplished using Seldon, TF Serving, Clipper and OpenVINO.

© 2020 Dell Inc. or its subsidiaries.

Business cases

To demonstrate how this platform will be used in a typical production environment, we leverage two real world business problems that are solved using machine learning. These two cases can be found as part of our common installation: price prediction for marketing (XGBoost) and visual classification application (TensorFlow). These two solutions are being provided as open source and are great starting point for a journey into using the proposed machine learning reference platform.

Link to Jupyter notebooks for use cases.

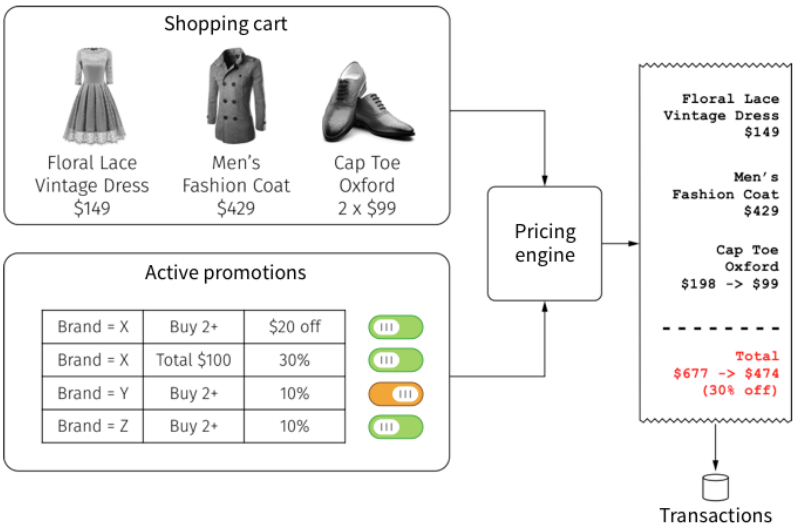
Example A: Marketing recommendation system

Stack: XGBoost, Kubeflow Python SDK, and Kubeflow pipeline
Target domain: Retail, marketing

This is a train-and-serve model that allows you to create simple prediction for marketing teams in two Jupyter Notebooks. It addresses the problem of finding optimal pricing for goods and products.

Pricing decisions are critically important for any business, as pricing is directly linked to demand and profits. Even one suboptimal decision can lead to tangible losses, and major mistakes can have grave consequences.

Although the fundamentals of price optimization are well understood, experience with leading retailers indicates that retail practitioners still struggle with pricing decisions. Even those who presumably make the right pricing decisions are often concerned they have unharvested profits or avoidable losses due to suboptimal pricing, and the lack of tools and techniques to measure it quantitatively.



© 2020 Dell Inc. or its subsidiaries.

Example B: “More like this” and visual search engines

Stack: TensorFlow, Keras, Kubeflow Python SDK and Kubeflow pipeline
Target domain: Retail, search engines, recommendation systems

This second example covers the more advanced use case of visual classification. The final model should recognize an uploaded photo of a dress and display two attributes, such as length and color.

This uses a pre-trained general-purpose model and a large data set of women’s dresses to train two neural network models: one to recognize color and the other to identify length. The TensorFlow framework and Intel Xeon powered machines help it achieve acceptable accuracy in a short training interval. Serving capability is part of Kubeflow and will be used to deploy this model as an API service. For this example, the web-based UI allows the user to upload a picture of a women’s dress and receive appropriate descriptions in a user-friendly manner.

black long

blue short

A few business cases adopted by our customers are similar to the visual classification use case described above. Three examples are provided below:

- Expanding product discovery with ML-powered image similarity search. The model can pick up the main characteristics of the sample dress and find similar-looking short striped dresses from the catalog. This feature can also be termed “visual recommendation” or “show me more like this.”
- A visual search for dinnerware patterns with Replacements.com. Finding a matching plate or mug is a challenging process for the customer. Often, they can’t remember the name of their pattern or the company that produced it. Grid Dynamics engaged with Replacements.com to design and train a set of deep learning models and develop a set of visual search APIs capable of matching patterns based on a photo uploaded by a customer.
- Identifying screws, a practical case study for visual search. Screws are visually very similar to each other, and therefore hard to distinguish. Traditional approaches to matching are frustrating and slow, and usually involve visual comparison to a huge catalog of choices. Wouldn’t it be nice if you could just point your smartphone to the screw that you need, and have it easily show up online?

© 2020 Dell Inc. or its subsidiaries.

Accelerating your data science journey

Even though Kubeflow is still under active development, Grid Dynamics has successfully implemented custom installations for its clients. As part of a new collaboration, Grid Dynamics is partnering with Dell Technologies and Intel to help customers implement and build data science platforms on Kubernetes by releasing reference architectures with Canonical and SUSE.

Dell and Grid Dynamics can provide customers comprehensive consulting services to accelerate their AI projects. These services include but are not limited to assessing our mutual customer's AI needs and providing them with right strategic solution and implementation of use cases. Dell and Grid Dynamics together now have the capability to not only deploy an entire data science infrastructure for their customers but also help them with custom modeling, algorithm and talent augmentation.

Dell and Grid Dynamics will provide technical support on their delivered solution. This collaboration allows customers to call for technical support from either of the companies and get collaborative support.

References

- Hidden Technical Debt in Machine Learning Systems
- Hadoop Distributed File System
- Kubernetes Architecture
- Declarative application management in Kubernetes
- Kubeflow
- Dell Reference Architectures
- SUSE Reference Architecture
- Canonical Reference Architecture

Learn more

- Grid Dynamics Data Science
- Dell and AI
- Intel and AI
- Intel Framework Optimizations
- Intel Builders
- Follow us on Twitter: #IntelBuilders
- Intel Xeon Scalable processors

CANONICAL

SUSE
We adapt. You succeed.

Copyright © 2020 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries.Other trademarks may be the property of their respective owners. Published in the USA in the USA 02/20 Whitepaper DELL-WP-AI-GRIDDYNAMICS-USLET-101.

Apache®, Beam®, Hadoop®, Kafka®, HBase®, Spark®, and Flink® are trademarks of the Apache Software Foundation. Canonical® and Ubuntu® are registered trademarks of Canonical Ltd. SUSE® and the SUSE logo are trademarks of SUSE IP Development Limited or its subsidiaries or affiliates. Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. Intel® and OpenVINO™ are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Grid Dynamics® is a trademark or registered trademark of Grid Dynamics International, Inc. in the United States and other countries. Kubernetes® is a registered trademark of The Linux Foundation. Google®, TensorFlow™, and Google Cloud Platform™ are trademarks of Google Inc. Amazon Web Services® is a trademark of Amazon Services LLC and/or its affiliates. Microsoft® and Azure® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The Jupyter® trademark is registered with the U.S. Patent & Trademark Office. Python® is a registered trademark of the Python Software Foundation. PyTorch is a trademark or registered trademark of PyTorch or PyTorch's licensors. Docker® is a trademark or registered trademark of Docker, Inc. in the United States and/or other countries.

Dell Technologies believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

DELL Technologies

intel