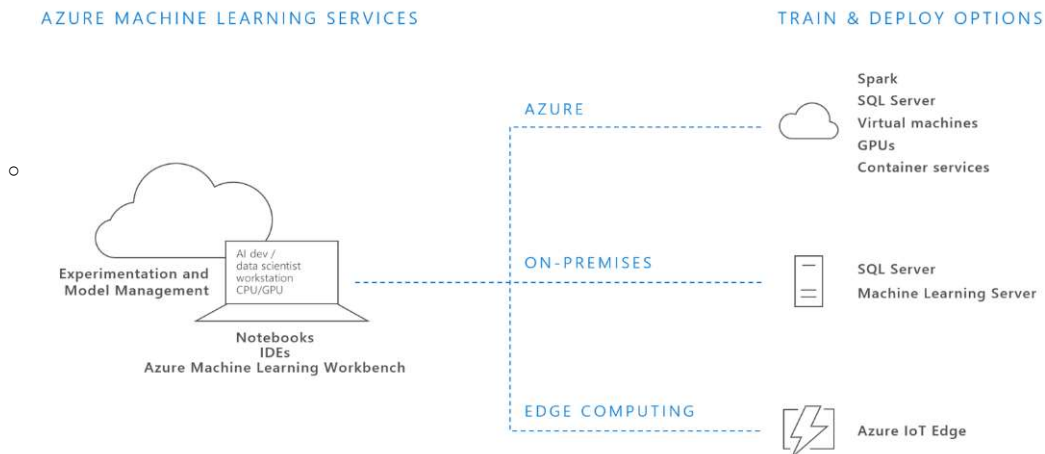


Documentation Notes

Monday, September 25, 2017 7:56 AM

- Overview - What is Azure ML?
 - Definition: integrated, end-to-end data science and advanced analytics solution. It enables data scientists to prepare data, develop experiments, and deploy models at cloud scale
 - Components
 - AML Workbench
 - AML Experimentation Service
 - AML Model Management Service
 - MMLSpark Library
 - Visual Studio Code Tools for AI

AZURE MACHINE LEARNING



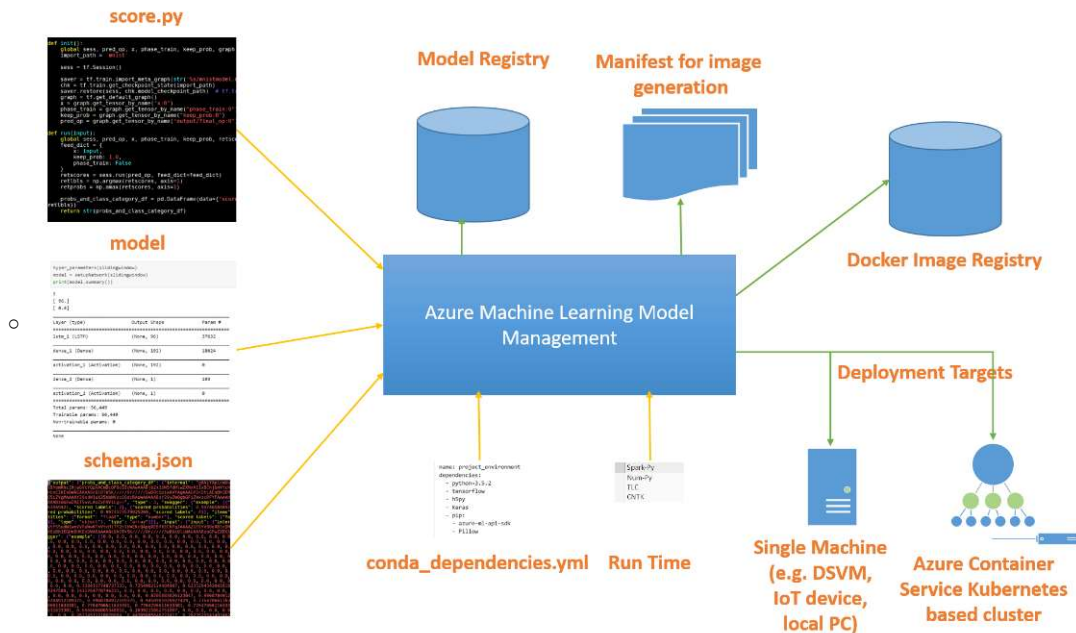
- Open source compatible - machine learning frameworks including scikit-learn, TensorFlow, CNTK, Spark ML, and tens of thousands more
- Open source compatible - AML is built on top of Jupyter Notebooks, Apache Spark, Docker, Kubernetes, Python, Conda, also MML Lib for Apache Spark and CNTK
- Microsoft ML technologies in AML - PROSE, microsoftml, revoscalepy
- AML Workbench: desktop application plus command line tools, supported on Windows and macOS. Takes you through the entire data science life cycle
- AML Experimentation Service: handles the execution of ML experiments. Also supports AMLW by providing project management, Git integration, access control, roaming and sharing
- AML Model Management Service: model versions and lineage are tracked from training runs to deployment. You can containerize your model, scoring scripts, and dependencies into Docker images.
- MMLSpark: open source Spark package that provides deep learning and data science tools for Apache Spark. It integrates Spark Machine Learning Pipelines with CNTK and OpenCV.
- Visual Studio Code Tools for AI: extension in Visual Studio Code to build, test and deploy deep learning and AI solutions. It has many integration points with AML
- What are the ML options from Microsoft? (Besides AML)

Option	Use	Integrates with	Value add
Microsoft ML Services in SQL Server	Run, train, and deploy machine learning models using R or Python, when you need to train or deploy models on-premises, or inside SQL Server	R, Python, AML Model Management	
Microsoft ML Server	When you need to build and deploy models built with R and Python on a server, distribute R and Python training at scale on a Hadoop or Spark cluster	Linux, Windows, Hadoop, Apache Spark, HDInsight, R Tools for Visual Studio, Python Tools for Visual Studio	Extends open source R and Python with support for various scenarios, provides proprietary packages that install with the server
DSVM	When you need to run or host your jobs on a single node, if you need to remotely scale up your processing on a single machine	Windows Server, Linux, AML Experimentation, AML Model Management	Supported as a target for both AML Experimentation and AML Model Management
▪ Spark MLLib in HDInsight	When you need to scale out your data processing and create models as part of a data pipeline	AML Experimentation, AML Model Management, AML Workbench	Spark can be used to scale out easily. You can author Spark jobs in Scala, Java, Python or R
Batch AI Training	Helps you experiment in parallel with your AI models and then trains them at scale. Describe your job requirements and configuration to run, and the rest is handled	AML Model Management, AML Experimentation (future)	Enables you to scale out deep learning jobs across clustered GPUs (using frameworks such as CNTK, Caffe, Chainer, TensorFlow)
CNTK	When you need to build a model using deep learning	All services	Easily realize and combine popular model types such as feed-forward DNNs, CNNs, and RNNs/LSTMs. It

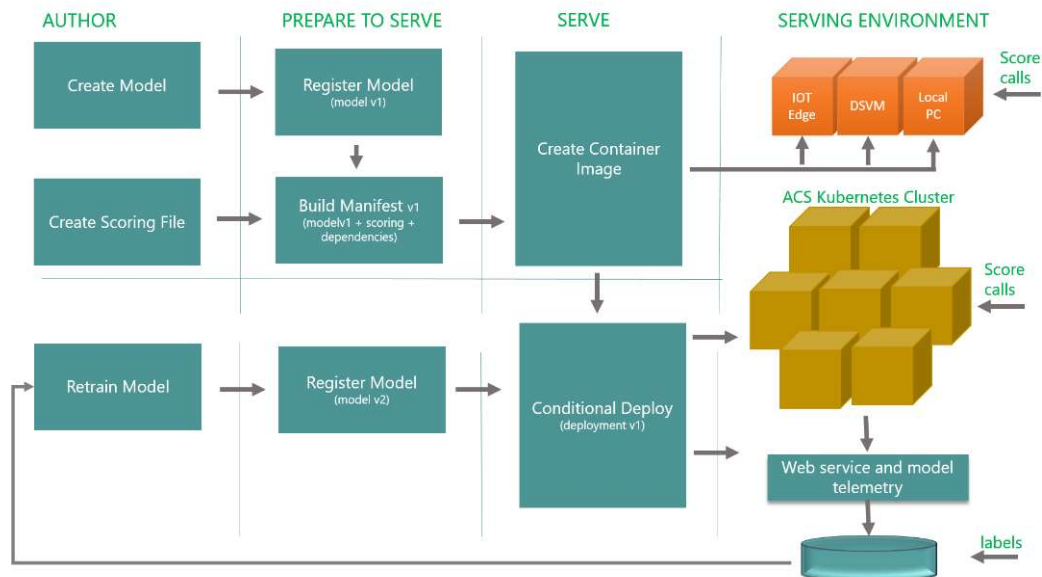
			implements SGD learning with automatic differentiation and parallelization across multiple GPUs and servers
Cognitive Services	When you need to build apps that use natural methods of communication	Across devices and platforms	30 APIs that allow your apps to see, hear, speak, understand, and interpret our needs with just a few lines of code

- Azure Machine Learning Model Management

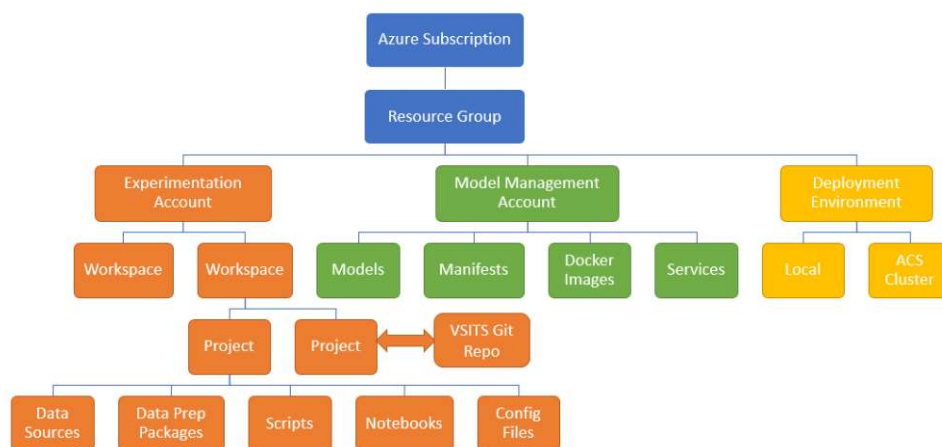
- Enables you to manage and deploy machine learning workflows and models
- Provides capabilities for model versioning, tracking models in production, deploying models to production through AzureML Compute Environment with Azure Container Service and Kubernetes, creating Docker containers with the models and testing them locally, automated model retraining, capturing model telemetry for actionable insights
- Provides capabilities through the CLI, API, and the Azure portal
- How models are registered and deployed into the cluster:



- Create and manage models
 - Tracking model versions in production
 - Captures all dependencies and associated information
 - Deeper insights? You can capture model telemetry using the SDK. Model telemetry is archived in user-provided storage.
- Create and manage manifests
 - Create a manifest that encompasses model, dependencies, inference script (aka scoring script), sample data, schema, etc. Acts as a recipe to create Docker container images.
 - Enterprises can auto-generate, create different, and manage manifests
 - Can use the manifests to build Docker-based container images that provides enterprises the flexibility to run these images on Kubernetes-based Azure Container Service, on-premises container services, development environments, and IoT devices
- Deploy Docker container images
 - With AML Model Management, you can deploy Docker-based container images with a single command to ACS managed by ML Compute Environment. These are created with a front-end server
 - Logging and telemetry by AppInsights
- Consumption: AML Model Management creates a REST API for the deployed model along with the swagger document. You can consume the APIs using a single CLI command, swagger-enabled applications, or using curl
- Retraining: AML Model Management provides APIs to retrain your models and update existing deployments with updated versions of the model.
 - You recreate the model in your experimentation environment
 - Then, you register the model with Model Management and update existing deployments
 - Updates performed using a single UPDATE CLI command - the UPDATE command updates existing deployments without changing the URL or the key



- [FAQ](#)
- [Create AML preview accounts and install AML Workbench](#)
- [Classifying Iris Tutorial Parts 1-3](#)
- [Bike-share tutorial: Advanced data prep with AML Workbench](#)
- Azure Machine Learning - Key Concepts



- Subscription: Workbench requires that each user has access to a valid Azure subscription and sufficient permissions to create resources.
- Experimentation account: you must have access to an experimentation account to use Workbench. It's an Azure resource (and billing vehicle) that contains your workspaces and projects, and you can add multiple users (seats) to an experimentation account.
- Model Management account: required Azure resource (and billing vehicle) for managing models
- Workspace: primary component for sharing and collaboration in AML, can be shared with multiple users that have been added to the experimentation account
- Project: logical container for all the work being done to solve a problem - maps to a single file folder on your local disk, you can add files/subfolders to it, and you can optionally associate it with a Git repository.
- Experiment: one or more source code file(s) that can be executed from a single entry point (may contain tasks like data ingestion, feature engineering, model training or evaluation). Currently, only Python/PySpark experiments are supported
- Model: the product of a ML experiment, recipes that generate predicted values.
- Compute target: the compute resource you configure for executing experiments (local computer, Docker container, Linux VM, HDInsight Spark cluster)
- Run: the lifetime of an experiment execution in a compute target. Information is captured and stored in the run history for each run within an experiment
- Environment: computing resource used for deploying and managing your models. Your model is hosted in a Docker container and exposed as a REST API endpoint
- Managed model: registered with an AML Model Management account
- Manifest: the recipe used to create the Docker container image (encompasses model, dependencies, scoring script, sample data and schema)
- Images: use manifests to generate/regenerate Docker images that create flexibility to run them in the cloud, locally, or on IoT devices
- Services: The webservice logic and dependencies are encapsulated into an image. Each webservice is a set of containers based on the image ready to service requests to a give URL. Model Management allows you to deploy models as web services