DataScience Platform with HDinsight

Alberto De Marco
Daniela Colombo
Wendy Frodyma
Kaushik Srinivasan
Jeroen Quakernaat

Microsoft

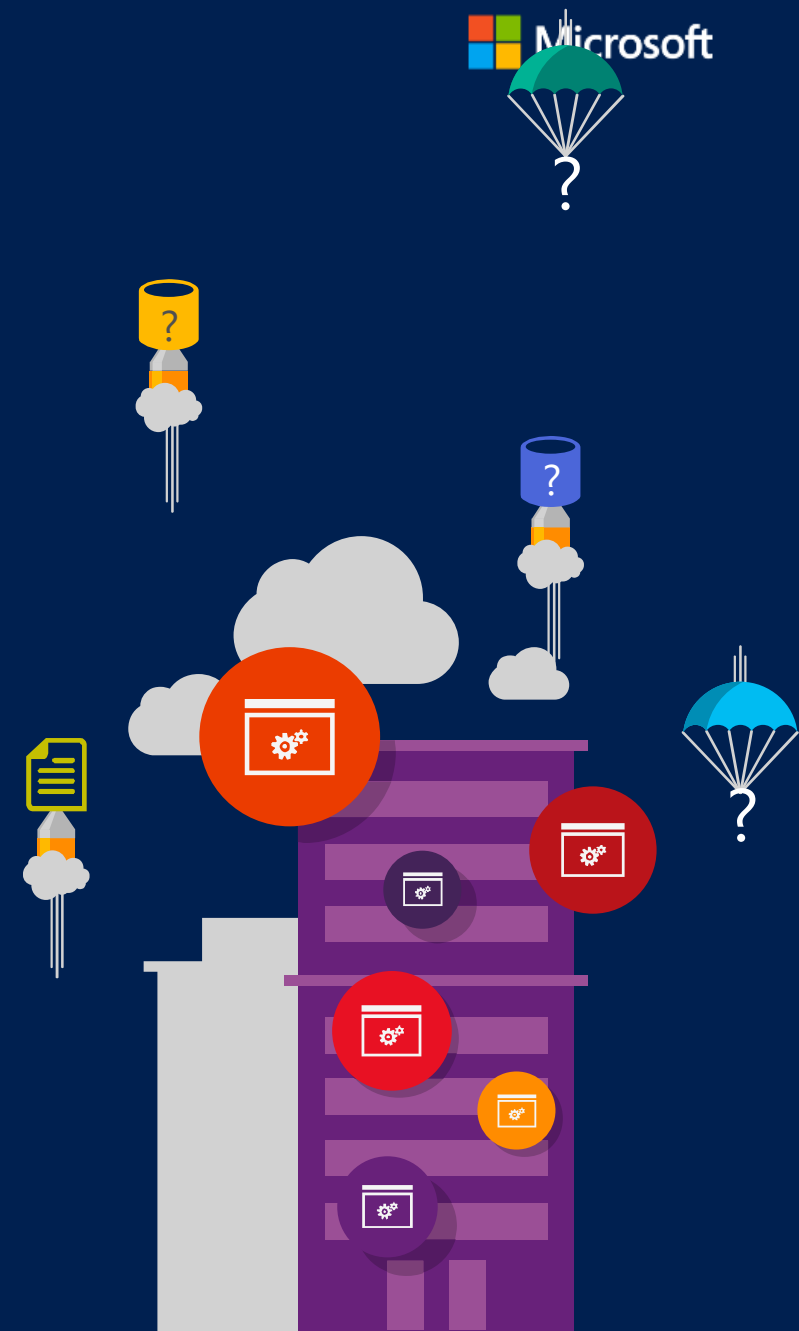
DS Team challenges to operatinalize

- IT provides to the Data Science Team a standard HDInsight cluster with Jupyter Notebooks & Azure Blob Storage as environment for their PySpark workloads.
- DS team reported these problems :
  1. Difficult productionizing process “re-inventing the wheel"
  2. ETL handover and maintenance struggles
  3. Not enough (parallel) processing power/memory
  4. No standard workflow, analysis sharing
  5. Difficult (unclear how) to add/share (new) data
  6. Unclear how to read/preprocess data, no best practices development
  7. Version control difficult without a constantly available server
  8. Difficult to use DS tooling

DS Team real pains

- Since they cannot add on HDInsight the python visualization libraries they need, they use mainly their laptops
- No structured way of sharing data between them (usb keys)
- Since they use their laptop each DS has his own environment and code is not always portable because needs libraries that are available only on the laptop of the DS.

Solution



How we addressed the pain points

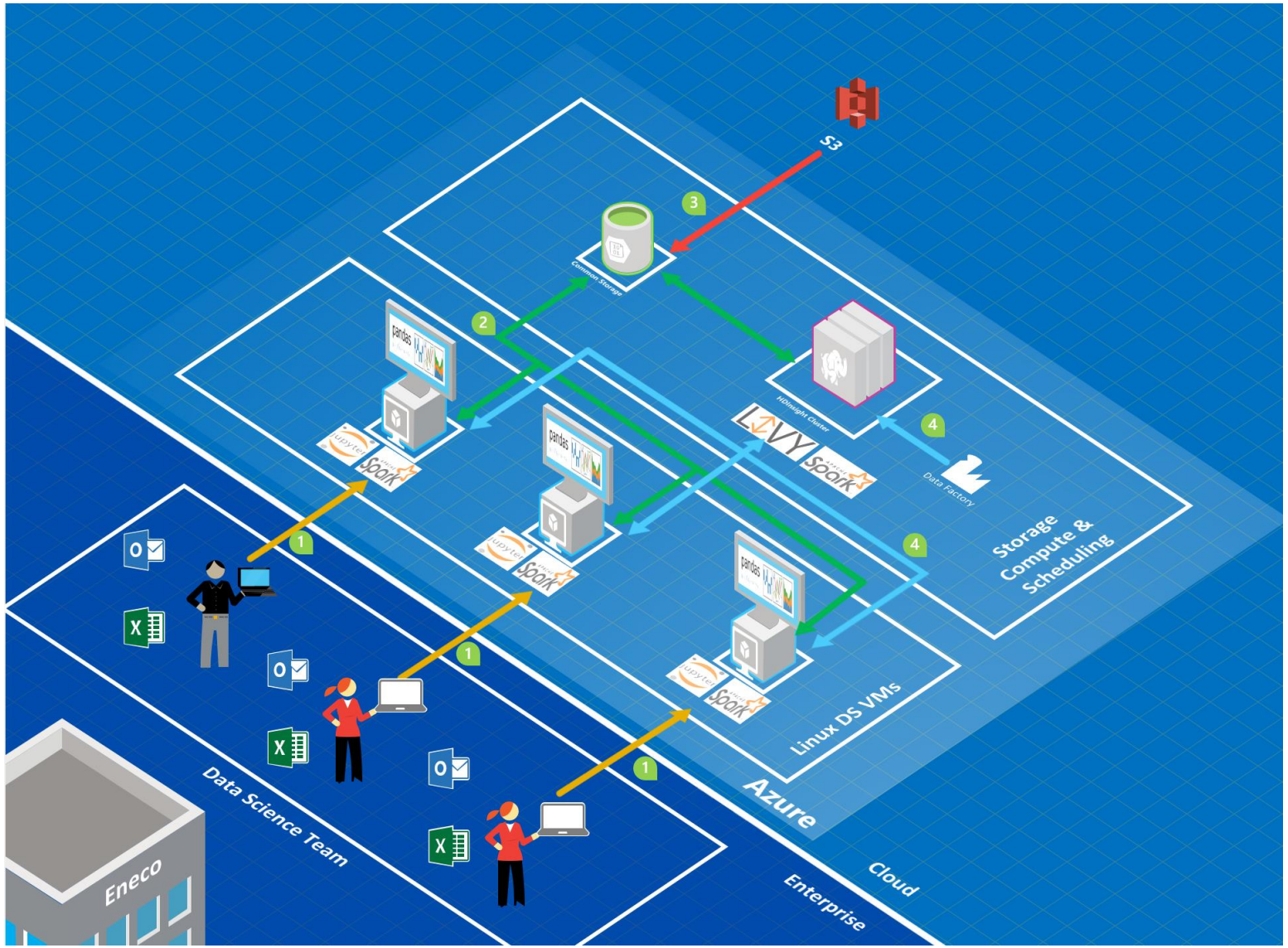
- Assign to each DS member a Linux DS VM with a local Jupiter notebook server and a local Spark installation (the OOB image that we offer from marketplace)
- Add to the OOB Linux DS VM the possibility to connect , via local spark, to azure blob storage (adding libraries, conf files and settings)
- Add to the OOB Linux DS VM spark magic (adding libraries, conf files and settings) to connect from local Jupyter notebook to the HDInsight cluster using Livy
- Tuning of the local spark parameters to overcome write speed bottlenecks on Azure Blob
- Leveraging , thanks to the ADF PG, the new Spark Activity on ADF to schedule PySpark jobs with custom modules loading
- Setup the HDInsight Azure Blob Storage as the “de facto” area where each DS finds input data and puts the outputs of his jobs with a clear and shared taxonomy and folder structure
- Setup of the git control version on the notebooks and python libraries on each DS VM

Results

- Common data area where all the DS collaborate together
- Structured and easy way to schedule Spark jobs
- Version control of notebooks (previously hosted on the cluster itself)
- Ability to run all the custom, needed external libraries on their own jupyter server on the linux DS VM

Our solution

1. Data scientist operate with standardized Linux DS VM
2. Transformed Data elaborated with Linux DS VM is read/written to Azure Blob Storage
3. New telemetry data from external sources comes to Azure Blob Storage
4. Heavy Spark computations are pushed/scheduled to the HDInsight Cluster that read/writes on the same storage blob



Thank you