Enterprise Strategy Group | **Getting to the bigger truth.**™

The Definitive Guide to Evaluating Cloud-based Apache Spark Platforms

Key Considerations and Best Practices from Selection to Proof of Concept

By Nik Rouda, ESG Senior Analyst & Vinny Choinski, Senior Lab Analyst

September 2016

Evaluating Cloud-based Apache Spark Platforms

Apache Spark is a powerful open source data processing engine built for speed, ease of use, and sophisticated analytics. This buyer's guide is intended to help decision makers evaluate available solutions and to understand the significant differences between them.

This guide is designed to help you:

1. Define evaluation criteria and compare common options.
2. Set up a successful proof of concept (PoC).
3. Estimate the total cost of ownership (TCO) and assess the return on investment (ROI).

This ESG Buyer's Guide was commissioned by Databricks and is distributed under license from ESG.

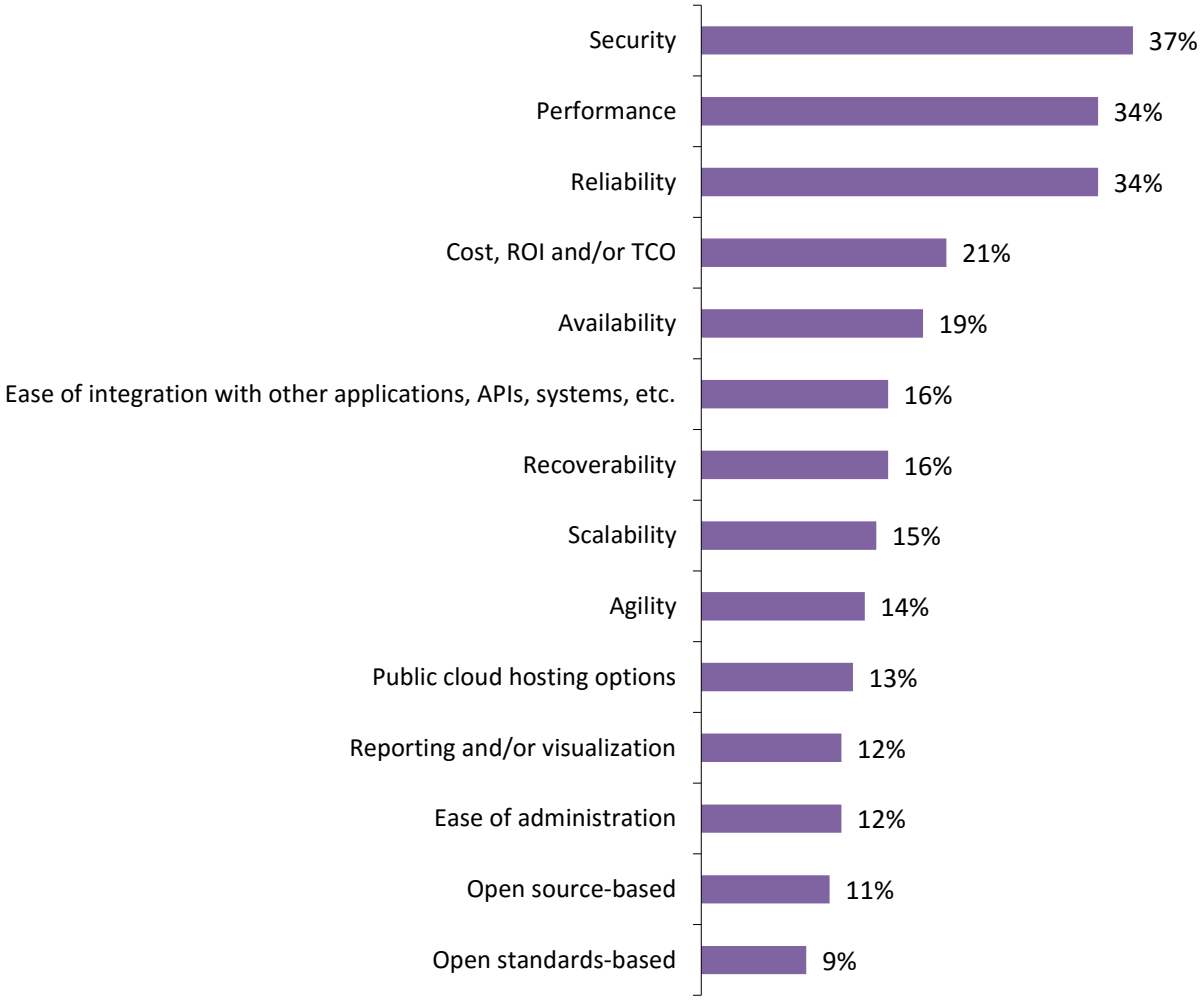
Section 1: Determine Your Evaluation Criteria and Compare Options

The first step in evaluating a Spark cloud platform is to fully understand what your buying criteria should be. ESG's research report, *Enterprise Big Data, Business Intelligence, and Analytics Trends: Redux*, found that organizations most often cited security, performance, reliability, and TCO as important attributes when considering any big data and analytics solution.[1]

FIGURE 1

Important Attributes for a Big Data and Analytics Solution

Which of the following attributes are most important to your organization when considering technology solutions in the area of big data and analytics? (Percent of respondents, N=475, three responses accepted)



1. Source: ESG Research Report, *Enterprise Big Data, Business Intelligence, and Analytics Trends: Redux*, July 2016.

These are good places to start, but two additional considerations in the context of Apache Spark you should evaluate are the level of domain expertise and quality of support the solution can provide. Here is a set of key considerations that you should keep in mind as you weigh the decision criteria.

Criteria	Considerations
High Performance at Scale	In addition to traditional measures of performance, it's also important to consider which option enables you to take full advantage of the flexibility of the cloud. Some key questions: • How has the vendor tuned open source Spark to optimize performance? • How easy is it to ETL data from disparate data sources? Does the vendor require you to lift and shift data? • How can users scale the platform to meet dynamic demand? Is it a manual process or is it automatic? • Does the platform allow you to scale compute independently from storage to achieve optimal performance?
Out-of-the-box Support for Diverse Teams and Use Cases	Apache Spark is a flexible engine with the potential to radically simplify your compute environment, if you can make it easy for the diverse teams throughout the organization to be productive on Spark. Some key questions: • How easy is it to stand up a notebook environment on the platform for data scientists and analysts? • Can they scale their notebook environment beyond a few individuals? Can they share their work and track changes? • How easy is it for developers and data engineers to put their Spark code into production? • How will DevOps or system reliability engineers monitor and troubleshoot the platform?
Deep Spark Expertise	Setting up, tuning, and scaling production-quality Spark applications is complex and requires levels of domain expertise that are in short supply. An excellent way to determine the level of Spark expertise that a vendor has and how that flows into the quality of support is to ask the following questions: • Does the vendor have committers that contribute back to the open source Spark project in-house? If so, how many do they have? • How significant have their contributions been to the open source Spark project? For example, are they editing documentation or leading the development of new algorithms to extend the use of the machine learning library (MLlib)? • Do they offer Spark training? How many individuals/organizations have they trained?

Efficient Cost Management	Building and deploying Spark applications involves more than just data engineering and data science. A critical part of a project involves building up the infrastructure to operationalize your Spark deployment, which can be costly. Key features that can greatly reduce TCO are:
	• Turnkey solutions, which reduce infrastructure cost by eliminating DevOps cost of cobbled-together systems.
	• Another way to reduce TCO is to better manage your compute costs as you scale to process more data. Here are some features that can lower operational costs:
	  • Auto-scaling - Are you able to scale compute instances by defining min or max number of instances to meet demand?
	  • Cluster sharing - Are you able to share a cluster across multiple applications and teams to reduce compute costs?
	  • Cluster reuse - Are you able to reuse clusters across jobs to reduce compute cost?
	  • Per-minute billing - Which vendors offer more fine-grained cost management with per-minute level billing of compute costs?

High Reliability of Service and Security of Data	As the usage of big data continues to increase, the need to securely provide "always-on" access to the data becomes more crucial for business success. Here are a few questions you can pose to ensure your organization can reliably and securely build advanced analytics solutions: • Does the platform have the ability to auto-recover failed clusters to reduce costly outages? • Does the platform offer granular role-based access controls across the entire big data stack to protect access to production compute resources? • Are you able to centrally employ identity and access management so permissions can be revoked easily in the event of an emergency? • Are you able to encrypt all derived metadata (notebooks, logs, and other artifacts) while at rest? Is data also secured during transit? • Does the platform keep an audit trail of all actions to meet compliance standards and monitor detailed usage patterns of the system as the business requires? • Does the platform offer cloud-native integration? Is it tuned and regression tested on the latest hardware from the cloud vendors?

Choosing the Right Data Platform

Once you've determined your buying criteria, you should identify available platform options and assess them based on those requirements. This section will guide you through the differences between the types of Spark cloud platforms that are available, and how to navigate even the subtlest differences in functionality, which can impact real-world outcomes.

There are a number of vendors for cloud-based Apache Spark platforms, and they fall within one of the following categories:

Hadoop Distribution Platform

Many Hadoop distribution vendors include Spark in their offering, but they generally do not excel at the software, services, and operating environment needed to run Spark in a cloud environment.

Key characteristics to evaluate:

- Their domain focus is Hadoop, which can directly impact quality of support due to lack of Spark expertise. Also, they tend to push their proprietary Hadoop tools over Spark, which will make your big data environment more disparate and difficult to manage.

- The DNA of these vendors is rooted in on-premises and professional services. They do not offer a cloud-native product, which can impact their ability to automatically provision resources, scale on demand, and continuously deliver patches, updates, and bug fixes.

- Beware of solutions that are not ready to work out-of-the-box because they will require heavy upfront investment in professional services before you can start being productive.

Infrastructure-as-a-Service Only

Many companies that want to do Spark believe that they have the internal resources and expertise to build and support their own implementation. If you fit that mold, you may want to consider a simple infrastructure management service to run and manage your clusters. The major cloud providers such as AWS usually offer an infrastructure management tool at a very low cost.

Key characteristics to evaluate:

- These tools are designed for infrastructure managers and DevOps, so they might not be built with Spark in mind. Integrating other components such as notebooks, security, and visualization tools will require some DevOps effort.

- Since these infrastructure management services are typically sold by the cloud providers themselves, it's not necessarily in their best interest to optimize performance because that would decrease the runtime of the cluster and reduce infrastructure spend.

- Be prepared to do a full cost analysis to understand how your plans to scale will impact your total cost of ownership. Some infrastructure solution providers do not clearly expose their pricing model, which is focused on driving up compute and storage costs. Not understanding the pricing strategy of each vendor may cost you in the long run.

Basic Managed Service

Basic managed services provide Spark in a hosted environment with minimal functionality for data science, engineering, and analysis. Since they lack the depth of functionality as a fully managed platform, they tend to offer a broader big data focus, with support for multiple technologies.

Key characteristics to evaluate:

- These services often offer Spark as one of many software packages (e.g., tools in the Hadoop ecosystem). This mixed bag of technologies can create needless complexity.

- If your big data stack includes multiple technologies, then this heterogeneous approach may be attractive, but keep in mind that Spark was designed to serve as a unified engine to eliminate these types of redundancies.

- They may not have the expertise to support Spark-focused organizations because they are focused on supporting a large ecosystem of tools. This can have major ramifications on the success of your project.

- The minimal functionalities provided may not easily scale out beyond a few individuals.

Fully Managed and Unified Platform

A fully managed platform provides comprehensive functionalities to support the workflows of diverse teams at large scale from end to end, including data ingest, exploration, production, and report creation.

Key characteristics to evaluate:

- Turnkey offerings should give you everything you need to start gleaning insights from your data on day one, which can greatly simplify operations and accelerate innovation.

- A single product can unify all analytic and production workloads—including SQL, machine learning, graph analysis, and stream processing. This eliminates the need to integrate best-of-breed tools that can become a DevOps nightmare.

- These platforms are considered premium offerings as they tend to contain everything you need to explore data, analyze insights, and put applications into production. Determine if they also have the expertise to support Spark.

Platform Selection

	Hadoop Platform	Infrastructure -as-a-Service	Basic Managed Service	Fully Managed Service
High Performance at Scale				
Data source connectors to simplify data access	✓	✓	✓	✓
Elastic scalability of on-demand clusters	✓	✓	✓	✓
Tuned clusters optimized for performance				✓
Deep Spark Expertise				
Spark training	✓		✓	✓
Expert Spark support				✓
Out-of-the-box Support for Diverse Teams and Use Cases				
Interactive notebooks with real-time collaboration and revision history				✓
Built-in visualizations and dashboards				✓
One-click deployment from notebooks to Spark jobs				✓
REST API access for cluster management and jobs	✓	✓	✓	✓
SSH access for debugging	✓	✓	✓	✓
Easy access to audit logs	✓	✓	✓	✓

Efficient Cost Management				
Turnkey platform with notebooks, job scheduler, cluster manager, security, etc.				✓
Cluster auto-scaling to meet demand			✓	✓
Auto-provisioning of clusters			Limited	✓
Sharing of clusters across teams and applications			Limited	✓
Cluster reuse across jobs to reduce compute costs				✓
Minute-level billing of compute costs				✓
Automatic migration between spot and on-demand instances				✓
High Reliability and Security				
Data encryption at rest and in flight	✓	✓	✓	✓
Audit logs for compliance and monitoring of usage patterns			✓	✓
Access control for clusters and notebooks				✓
Fault-tolerant clusters				✓
Permission-based job and workflow execution				✓

ESG Recommendations for Selection of an Apache Spark Platform:

If you are focused on using Spark for all your workloads, our advice is to choose a fully managed service from a vendor that has the expertise and support you need. Performance at scale facilitates the ability to look broader and deeper in larger data sets, and leads to more accurate and nuanced insights. Spark expertise can solve challenges faster, enabling you to be more productive in your analytics. Facilitating diverse teams will bring together different ideas and viewpoints, and generate more meaningful understanding. Cost management reduces wasted spending, allowing you to focus resources on innovation, not administration. Reliability and security deliver the quality and confidence needed for the business to go into production with analytics applications, safely.

Several basic cloud infrastructure services are available, but they are inherently limited in their capabilities and force you to build your own operating functionality. ESG recommends instead comparing fully managed services on the market, and evaluating Databricks as a high-profile vendor in the category. Databricks was founded by the creators of Spark and as such, it is best suited to support the technology today and as it continues to rapidly evolve in the future. Databricks provides a solution that is ready to run with everything your data scientists, data engineers, and data analysts will need to be productive right away. Further, its Spark platform encapsulates many product innovations designed to reduce TCO and provide a highly reliable and secure operating environment.

Some still feel they must pursue a "best-of-breeds" approach and assemble their own customized Spark environment, bringing additional complexity to their big data infrastructure. Even in this case, Databricks can still serve as a powerful Spark solution alongside a general purpose analytics platform for other workloads.

Section 2. Set Up a Successful PoC

While it is important to understand the criteria mentioned, it is always worthwhile to test a solution for yourself before making a long-term commitment. This will really show the nuances and differences of your approach most clearly.

Presented here are the recommended phases for an evaluation, whether you call it a test, proof of concept (PoC), or full pilot. This section is designed as a worksheet for building your detailed test plan; it is not the complete specification of the tests themselves. Every business will have its own particular demands. Please use this as a starting point and tailor as needed to fit your environment.

High-level PoC Test Plan

Kickoff, Setup, and Responsibilities	What to Look for
Scoping, Use Case, and Design	Work with the vendor to identify and define use cases and key evaluation criteria, and then design accordingly.
Onboarding	Work with vendor to schedule a training/workshop to learn the basics of Spark and get up to speed quickly on the vendor solution.
Cross-team Evaluation	Ensure you have representatives from the various teams participating in the POC (i.e., data science, engineering, analysts, etc.)
Scheduling	Bound the time to achieve PoC goals. Schedule intermediary objectives to ensure the PoC stays on track. Note the time it takes for your team to get started with the PoC as that is a good indicator of the difficulty of adapting the solution at scale.
Preparation	Prepare a test data set for evaluation of the vendor solution.

Product Evaluation	What to Look for
General Data Analytics	
Evaluate the basic features in the solution and how it handles ad hoc data exploration.	
	Data access and ingestion: How easy is it to connect to disparate data sources? Does the platform support schema-on read?
	ETL and cleansing: How much time is spent on ETL and data preparation?
	Data exploration: How easy is it to get the notebook environment up and running? Does the environment support all the programming languages used by your team (e.g., R, Python, Scala, and SQL)?
	Sharing and collaboration: How easy is it for multiple people to work on the data together? Do you have the ability to comment in real time, view logs, and access version history?
	Visualization and reporting: Are there built-in visualizations? Can you integrate with common BI tools? How does the tool help you create dashboards?
	Self-service cluster management: How easy is it to provision clusters without DevOps support? Are the clusters fault tolerant? Are you able to provision clusters based on demand? Are there cost management features such as auto-scaling, cluster sharing, spot instances, and cluster reuse?
	Backward compatibility: Does the platform allow you to run multiple Spark versions simultaneously?
Advanced Analytics	
Evaluate the more advanced features in the solution and start building more sophisticated use cases such as machine learning or a more complicated data warehousing use case.	
	Model building: How rapidly can you build and train machine learning models?
	Pipeline generation and featurization: Are you able to extract features from the data at scale?
	Production: How do you deploy algorithms into production?
	Library integration: How does the platform help you to manage the third-party libraries your team relies on?
Real-time Streaming	
Evaluate whether the solution can support streaming/continuous application use cases.	
	Real-time ingest: Can the platform handle large volumes of streaming data in real time?
	Streaming ETL/analytics: Are you able to process data streams in real time?
	High-availability deployment: Does the platform offer fault tolerance and auto-recovery features to ensure high availability?
Production	
Evaluate the production, debugging, and extensibility capabilities of the platform.	
	Debugging: Is the Spark web UI integrated into the platform for monitoring your application and tracking job progress? How easy is it to access audit logs?
	Production jobs: Can you schedule your jobs directly from the platform? Can you monitor and receive alerts?
	Programmatic access: Does the platform offer fine-grained controls such as RESTful APIs, SSH access, and init scripts?

Post POC	What to Look for
Partner	Partner with the vendor to co-present the findings and results to key stakeholders and decision makers.

Section 3. Estimating the Total Cost of Ownership

Once you've evaluated various solutions and completed your PoC, you will need to determine what your ROI will be. To estimate your TCO, you should capture both the CapEx and OpEx comprehensively, including:

- Cost and time to build and get a system up and running.
- Cost of compute resources over time and comparison of how efficient each platform is at resource utilization.

Engineering Costs

As previously covered in the evaluation criteria section, the cost to operationalize a cloud-based Spark platform can vary by vendor. It's important to estimate how much work and time it would take to:

- **Set up the infrastructure** by developing necessary features such as single sign-on, integrating third-party tools like a notebook solution or BI tool, setting up version control, and more.
- **Maintain the system** once it's up and running. This could include managing Spark upgrades, adding capacity, monitoring the clusters, troubleshooting and debugging, and other DevOps-related work to ensure availability and performance.
- **Provide ongoing Spark support.** If you don't have in-house expertise, this can be very costly, as you need to consider training the team on Spark, fixing issues like slow jobs, troubleshooting Spark package installs, and other Spark support topics.

See our *Engineering Cost Estimator Worksheet* in the Appendix to help you estimate your engineering costs.

Speed to Deployment

- The quicker you can stand up your Spark cloud platform, the faster you can start realizing value from your data. You want to look at how the platform helps reduce the work effort needed from an engineering perspective. Reducing the time spent on setup, maintenance, and ongoing support will allow you to avoid delays and downtime so you can get to your end goal faster and more efficiently.

- To quantify the potential impact a particular Spark cloud platform can have on your ability to successfully deploy your project in a timely manner, simply sum up the total work effort (weeks) needed for each section (infrastructure, DevOps, and Spark support) in the Engineering Cost Estimator Worksheet.

Compute Costs

- As you scale, your consumption of compute resources will also grow, resulting in a higher usage bill at the end of the month. It's important to understand how many nodes and node hours per day you'll need by looking at historical data and daily averages over time.

- Once you have your estimated baseline of compute costs, you can estimate the value of the efficiency gains provided by product innovations such as access to the latest and fastest version of Spark, expert support to help optimize workloads, and cluster management features designed to reduce compute costs such as those associated with auto-scaling clusters, the use of spot instances, and cluster reuse.

Total Value of Platform

- Once you've measured your engineering costs, the impact delays and downtime can have on your deployment, and compute costs at scale, you'll have a clearer picture of your TCO and the value-add that can be realized by your business.

The Bigger Truth

You should be deliberate and thoughtful in your strategy when designing your Apache Spark initiative. A poor decision will prove very costly in the long term, cause significant delays in business outcomes, and jeopardize future capabilities. You should define your functional needs not just for the data scientist and related specializations, but also for IT infrastructure and operations, developers, and the various lines of business activities and all their diverse use cases.

In your evaluation, wherever possible, consider and measure not just the ability to "get it done" but also "how hard it is." Some products may seem simpler, but offer less, which will ultimately put more of the burden on you to fill in the gaps and support your workarounds. This ongoing effort can be easily underestimated.

Guiding principles for your evaluation of a Spark platform should include:

- **Faster Innovation** - The key to success is shortening and simplifying that path from data to end results as much as possible. The platform shouldn't add complexity.

- **Spark Expertise** - If you're betting on Spark, you should work with vendors that have deep expertise to ensure success. It's never a good idea to invest in a vendor that doesn't have the in-house skills to solve complex Spark problems.

- **High Reliability and Security** - It is critical to ensuring clusters are always up and running (to avoid outages that can mean revenue loss) and the data is secure at all times (think sensitive customer data).

- **Managing Costs and ROI** - Just because a solution is the cheapest of the bunch upfront, doesn't necessarily mean it will be the cheapest in the long run. Consider upfront capital costs along with ongoing operational costs and any efficiency gains delivered by the product.

Most importantly, focus again on your overall company goals. Do you want to build and manage your own Spark environment or leverage the best possible choice on the market? Find a solution you can use as an effective tool for the real work of getting business value from big data analytics.

Engineering Cost Estimator Worksheet		
Use this worksheet to estimate the engineering costs needed to set up a Spark solution.		
Cost Per FTE: The average hourly rate of your FTE		
	Work Effort Required (Weeks)	Total Costs (Number of Weeks x Cost Per FTE)
Infrastructure Development: When selecting a non-turnkey solution, you will need to consider developing certain capabilities in-house and integrating other tools to meet your requirements.		
- Implementing login controls		
- Installing a notebook solution for exploration		
- Setting up and maintaining a production job scheduler		
- Setting up external libraries		
- Setting up version control		
- Integrating external BI tools		
- Other feature development and integrations		
DevOps Maintenance: Once you have your Spark solution up and running, you'll need to consider the operational costs to maintain the system.		
- Managing Spark upgrades, patches, and bug fixes		
- Provisioning clusters		
- Monitoring clusters		
- Troubleshooting and debugging issues		
- Other DevOps maintenance		
Spark Support: Keep in mind the level of support you'll need from the vendor. If you lack the Spark expertise, this can also have a major impact on team productivity.		
- Training new employees on Spark		
- Tuning Spark for performance		
- Troubleshooting and debugging issues		
- Connecting third-party applications		
- Other Spark support		
Summary		
Feature Development Costs		
DevOps Maintenance Costs		
Spark Support Costs		
Total Engineering Cost		