DUMPSBOSS.

IBM Big Data Architect

IBM C2090-102

Version Demo

Total Demo Questions: 13

Total Premium Questions: 130

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Big Data Fundamentals	24
Topic 2, Big Data Architecture	32
Topic 3, IBM Big Data Platform	46
Topic 4, Big Data Solution Design	28
Total	130

QUESTION NO: 1

Which of the following practices should be included when designing for high availability of a Hadoop platform? (Choose two.)

A. Deploy active and standby NameNodes

B. Store every replica of a block on one rack

C. Distribute DataNode replicas across racks

D. Run all cluster services on a single management host

E. Disable DataNode heartbeats

ANSWER: A C

Explanation:

Deploy active and standby NameNodes and **distribute DataNode replicas across racks** are appropriate high-availability practices. NameNode high availability removes the single point of failure for HDFS namespace services by maintaining an active NameNode and a standby NameNode that can take over when required. The standby keeps its namespace state synchronized through shared edit-log mechanisms.

Rack-aware replica placement protects data availability against failures that affect an entire rack, such as a top-of-rack switch or power problem. HDFS places replicas across failure domains so that a single node or rack outage is less likely to make data unavailable. The Apache Hadoop HDFS High Availability Guide and the HDFS Architecture Guide describe these capabilities.

QUESTION NO: 2

Which of the following are characteristics of a logical data warehouse architecture? (Choose two.)

A. Data can remain in multiple physical repositories

B. Users can access data through a common logical access layer

C. Every data source must be migrated into one physical database

D. Metadata and governance are no longer required

E. Only unstructured data can be included

ANSWER: A B

Explanation:

Data can remain in multiple physical repositories and **users can access data through a common logical access layer** are characteristics of a logical data warehouse. This architecture recognizes that data may be retained in different platforms according to workload, cost, latency, or governance needs. A logical integration layer can provide a more unified access experience without requiring all data to be copied into one physical warehouse.

A logical data warehouse does not require every source to be replaced, nor does it eliminate the need for governance, metadata, and data-quality controls. IBM describes data virtualization as providing access to distributed data without physically moving it, which is a key principle used in logical data warehouse architectures. See IBM's overview of data virtualization.

QUESTION NO: 3

What are the two levels documented in the Operational Model? (Choose two.)

A. Logical

B. Rational

C. Theoretical

D. Physical

E. Middleware

ANSWER: A D

Explanation:

The Operational Model is documented at the **Logical** and **Physical** levels. The logical level describes the required operational capabilities and the relationships among them without committing the design to particular products, servers, or deployment topology. It gives architects a technology-independent view of how the platform must be operated, managed, secured, monitored, and supported to meet business and service requirements. The physical level then realizes that logical design in an implementable environment. It maps the operational capabilities to actual infrastructure, software components, network placement, resource sizing, and deployment decisions. Using both levels lets an architecture remain understandable and stable at the conceptual design stage while still providing the implementation detail needed for delivery and operations. This separation is consistent with IBM architecture practices, which distinguish logical architecture from its physical realization and deployment. IBM's big-data architecture guidance is available in the IBM Redbooks publication, IBM Big Data Platform: Bringing Big Data to the Enterprise; IBM also provides broader architecture resources through its IBM Architecture Center.

QUESTION NO: 4

Which of the following statements are TRUE regarding Apache Spark? (Choose two.)

A. Spark can run applications on YARN

B. Spark supports in-memory processing for iterative workloads

C. Spark requires every data set to be stored in a relational database

D. Spark can only process streaming data

E. Spark replaces HDFS block replication

ANSWER: A B

Explanation:

Spark can run applications on YARN and **Spark supports in-memory processing for iterative workloads**. YARN is a supported Spark cluster manager, allowing Spark applications to share Hadoop cluster resources with other YARN workloads. Spark also persists data sets in memory when appropriate, which can improve performance for iterative algorithms and repeated access to the same intermediate data.

Spark does not require all data to reside only in a relational database, and it is not limited to streaming workloads. It provides a general distributed-processing engine with libraries for SQL, streaming, machine learning, and graph processing. See the Apache Spark documentation and the Spark documentation for running on YARN.

QUESTION NO: 5

Which of the following is the artifact that assists in ensuring that the project is on the right path toward success?

A. Component Model

B. Empathy Map

C. Viability Assessment

D. Opportunity Plan

ANSWER: C

Explanation:

Viability Assessment is correct because it provides a structured, evidence-based check that the proposed initiative can realistically deliver the intended business outcome. It assesses whether the work remains practical and worthwhile by considering factors such as expected value, stakeholder and market fit, delivery feasibility, costs, risks, constraints, and the assumptions on which the initiative depends. Teams use its findings to validate their direction, identify issues early, refine scope or priorities, and make informed decisions about whether to continue investing in the work. This makes it a useful project artifact for maintaining alignment between the solution being developed and the conditions required for successful delivery and adoption. In an iterative delivery approach, viability is not merely an initial approval activity; it should be revisited as the team gains evidence from users, implementation work, and changing business conditions. IBM Garage emphasizes continuous collaboration, experimentation, and evidence-driven decisions while moving an idea toward a scalable solution. IBM also describes enterprise design practices that focus teams on user outcomes and measurable value. See IBM Garage and IBM Enterprise Design Thinking.

QUESTION NO: 6

Which of the following statements regarding Big R is TRUE?

A. Unless specified otherwise, Big R automatically assumes all data to be integers

B. Big R's 'bigr.frame' is equivalent to R's 'data.frames'

C. When you execute Big R "apply" function, Big R transparently extracts data out of HDFS into the Big R engine

D. A data analyst using Big R employs MapReduce programming principles

ANSWER: A

Explanation:

Unless specified otherwise, Big R automatically assumes all data to be integers. Big R was designed to let R users analyze data stored and processed in Hadoop without requiring them to write MapReduce programs directly. Because the data representation and processing model must be suitable for distributed execution, Big R uses defined assumptions when interpreting data unless the analyst supplies explicit metadata or type information. The default integer assumption is therefore important when preparing a `bigr.frame` and related Big R input: analysts should explicitly declare the intended column types when their source contains other types, such as character strings or floating-point values, so that the distributed computation interprets values appropriately. This behavior reflects Big R's Hadoop-oriented data model rather than ordinary in-memory R object handling. IBM's BigInsights materials describe Big R as an R analytics capability integrated with Hadoop, with distributed data structures and operations intended for large-scale analysis. See IBM's InfoSphere BigInsights Redbooks publication and the IBM BigInsights documentation for the Big R and BigInsights platform context.

QUESTION NO: 7

Which of the following is NOT an example of unstructured data?

A. HBase table

B. Tweet

C. Netezza table

D. Internet Protocol Detail Record

ANSWER: C

Explanation:

Netezza table is the correct answer because it is a relational database table and therefore stores data in a defined, organized schema. A Netezza system is designed for high-performance analytics on structured relational data: tables have explicitly defined columns, data types, rows, and metadata, and they can be queried using SQL. This predictable tabular organization is the defining characteristic of structured data. For example, a Netezza table may contain a customer identifier, transaction date, product code, and amount, with each value placed in a specified column and conforming to its declared type. That design enables consistent validation, indexing or distribution choices, joins, aggregations, and governed reporting. IBM describes structured data as data organized according to a predefined model, commonly in rows and columns, which makes it readily searchable and analyzable with relational tools. Netezza provides exactly this relational-table model, so it is not an example of unstructured data. See IBM's overview of structured versus unstructured data and IBM documentation for Netezza tables.

QUESTION NO: 8

You need to set up a distributed storage system for being able to process very large data sets and you want to be able to leverage the Open Data Platform (ODP) Core. Which one of the following would you use?

A. Apache Spark

B. IBM GPFS

C. EMC Isilon

D. HDFS

ANSWER: D

Explanation:

HDFS is the appropriate choice because the Hadoop Distributed File System is the distributed storage layer associated with the Apache Hadoop ecosystem that underpins the Open Data Platform Core. It is designed to store very large files across a cluster of commodity servers, dividing files into blocks and distributing those blocks among DataNodes. This approach provides aggregate storage capacity and throughput beyond what a single server can supply. HDFS also maintains replicas of data blocks, enabling the system to continue serving data when individual nodes or disks fail. Its architecture is specifically optimized for high-throughput, streaming access to large datasets, which is the common storage pattern for Hadoop-based analytics and batch processing workloads. The Open Data Platform Core established a common, compatible set of Hadoop components and interfaces; HDFS is the core distributed file-system component in that stack. Workloads can therefore use HDFS as shared cluster storage while Hadoop processing services access the data where it resides. Apache's HDFS architecture documentation describes its block-based, replicated distributed design and its suitability for large-scale data processing: Apache Hadoop HDFS Architecture. The Open Data Platform's Core specification is described by the Open Data Platform Initiative.

QUESTION NO: 9

Which of the following statements are TRUE regarding HDFS rack awareness? (Choose two.)

A. It helps place replicas across multiple racks

B. It improves resilience to rack-level failures

C. It removes the need for HDFS block replication

D. It requires each replica to be stored on a different rack

E. It replaces YARN resource scheduling

ANSWER: A B

Explanation:

It helps place replicas across multiple racks and **it improves resilience to rack-level failures** are true. HDFS uses topology information to make replica-placement decisions. Placing replicas on different racks reduces the likelihood that a single rack, switch, or power-domain failure removes all accessible copies of a block. Rack awareness also helps Hadoop make data-locality decisions when scheduling processing tasks.

Rack awareness does not eliminate the need for block replication, and it does not mean every replica is necessarily stored on a different rack. HDFS replication policy balances fault tolerance, network traffic, and storage utilization. See the Apache Hadoop HDFS Architecture Guide.

QUESTION NO: 10

A bank wants to build a system that tracks all ATM and online transactions in real-time. They want to build a personalized model of their customer's financial activity by incorporating enterprise data as well as social media data. The system must be able to learn and adapt over a period of time. These personalized models will be used for real time promotions as well as for any fraud or crime detections. Given these requirements, which of the following would recommend?

A. Spark

B. Hadoop

C. Cloudand

D. Netezza

ANSWER: A

Explanation:

Spark is the appropriate recommendation because it provides a unified distributed processing platform for both continuously arriving transaction events and the large historical data sets needed to build customer models. Spark Structured Streaming supports scalable, fault-tolerant stream processing and can apply the same DataFrame and SQL-based transformations used for batch data to live ATM, online-banking, and social-media feeds. This makes it practical to enrich incoming transactions with enterprise customer and account data, calculate features, and produce low-latency promotion or fraud-scoring outputs.

Spark also includes MLlib, which provides distributed machine-learning algorithms and pipeline capabilities for preparing data, training models, evaluating them, and applying fitted models at scale. Historical and newly accumulated behavior can therefore be used to retrain and refine personalized models over time. Its in-memory execution model is particularly useful where scores must be generated quickly enough to support real-time intervention or offers. In an IBM environment, Spark workloads can be deployed and managed through IBM Analytics Engine, which is designed to run Apache Spark applications. See the Apache Spark Structured Streaming documentation and Spark MLlib guide.

QUESTION NO: 11

A major telecommunication company has millions of customers. Most of their customers are prepaid. Being prepaid customers, they can very easily switch to other vendors. The last four to six months, this company has lost quite a good number of customers to competition. They intend to build a system that can provide them with insight into the customer's social network (e.g. who is the influencer and who is the follower). They also want the ability to monitor the voice and data usage patterns in real time and they want the system to be trained over time to predict possible dissatisfactions. Given this scenario, which one of the following would you recommend?

A. Hadoop

B. Spark

C. Cloudant

D. Netezza

ANSWER: B

Explanation:

Spark is the appropriate recommendation because the proposed solution requires a unified analytics platform for streaming data, graph-oriented relationship analysis, and iterative machine-learning workloads. Spark Structured Streaming can continuously process incoming events, such as voice-usage and data-usage records, with scalable distributed processing and stateful operations. This enables near-real-time feature creation and alerting for behavior patterns associated with likely churn. See the Apache Spark Structured Streaming documentation for its processing model and streaming capabilities.

The social-network requirement maps well to Spark's graph-processing capabilities. Customer relationships can be represented as vertices and interactions as edges, allowing graph algorithms to identify influential customers, connected communities, and follower-like relationship patterns. Spark also supports machine-learning pipelines that can repeatedly train and apply churn or dissatisfaction prediction models as new labeled outcomes become available. Its in-memory, distributed execution is particularly useful for iterative model training and graph computations across a very large prepaid subscriber base. Apache Spark documents its graph abstraction and graph algorithms in the GraphX programming guide, while its MLlib components provide scalable tools for building predictive analytics workflows.

QUESTION NO: 12

A Hadoop cluster will store sensitive customer data. Which of the following controls should be included in the security architecture? (Choose two.)

A. Kerberos authentication

B. Authorization policies for HDFS and related services

C. Shared administrator credentials for all users

D. Disable audit logging to improve performance

E. Anonymous write access to ingestion directories

ANSWER: A B

Explanation:

Kerberos authentication and **authorization policies for HDFS and related services** should be included. Kerberos provides strong authentication for Hadoop users and services, helping the cluster establish the identity of a requesting principal. Authorization then determines whether that authenticated identity can read, write, administer, or otherwise use protected data and services. Hadoop supports permissions and access-control mechanisms for HDFS and ecosystem services.

Disabling audit logging or using shared administrator credentials weakens accountability and makes investigation more difficult. Security architecture should also address encryption, auditing, key management, and least-privilege administration where required. See the Apache Hadoop Secure Mode documentation and the HDFS Permissions Guide.

QUESTION NO: 13

A reputable market research firm wants to explore more business opportunities. They have great in house skill in python and machine learning. Their business model is simple, they build the solutions for customers using python and machine learning algorithms and give these solutions to the customer's engineering team for implementation. Given this scenario, which of the following would you recommend?

A. Netezza

B. Spark

C. Cloudant

D. Hadoop

ANSWER: B

Explanation:

Spark is the best recommendation because it directly aligns with a team whose existing strengths are Python development and machine learning. Spark provides PySpark, a first-class Python interface for distributed data processing, allowing the firm to develop data preparation, feature engineering, model training, and scoring workloads in a familiar language. Its MLlib and the higher-level *pyspark.ml* APIs provide reusable machine-learning pipelines, including transformers, estimators, evaluation, and tuning capabilities. That structure is particularly useful when delivering a solution to a customer engineering team: the team can receive repeatable pipeline code rather than a one-off analytical result.

Spark is also designed to scale the same Python-based solution from local development or small datasets to distributed execution on a cluster. This supports the firm's consulting model, where customer data volumes and deployment environments may differ substantially. Spark can run independently or alongside common big-data platforms, giving customer engineering teams flexibility in operational implementation while retaining the delivered Python and machine-learning logic. Apache Spark documents its Python API and its scalable machine-learning library at PySpark API documentation and Spark MLlib guide.