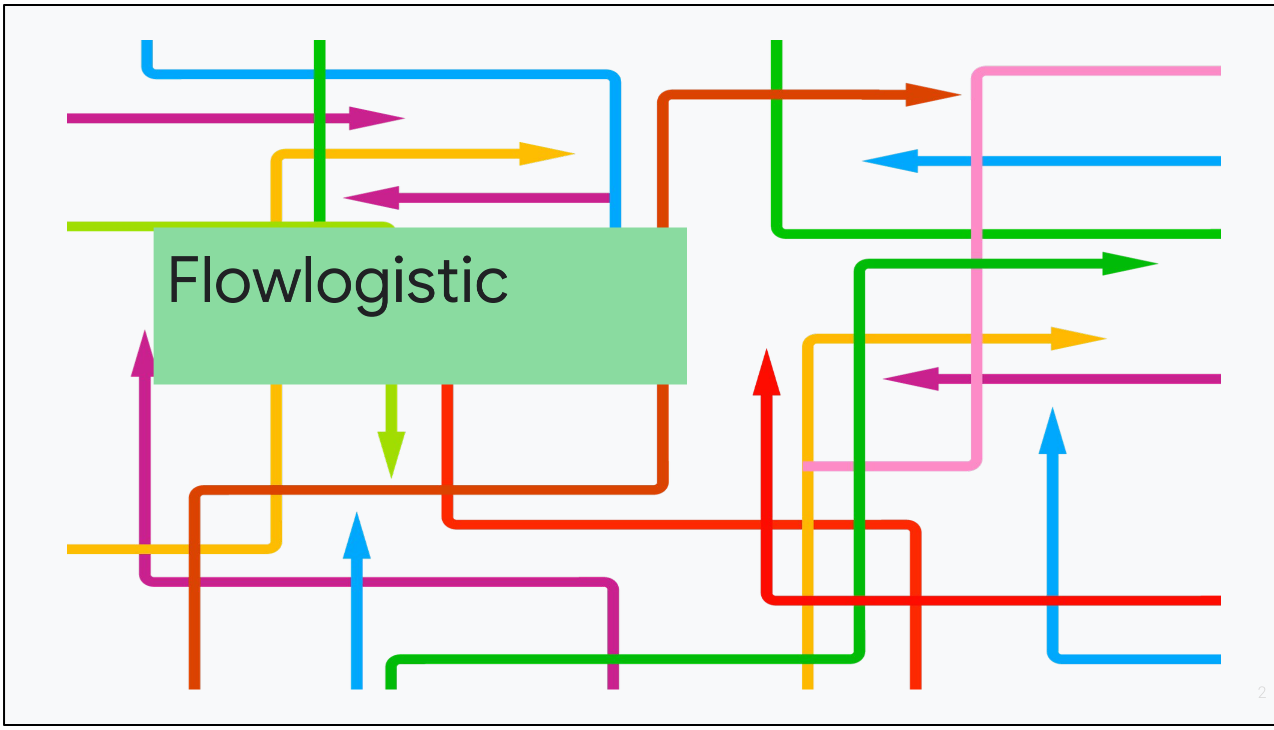
Sample Case Studies for the Professional Data Engineer Exam



The case is no longer provided by the Certification organization, but is now offered in this course for training purposes.

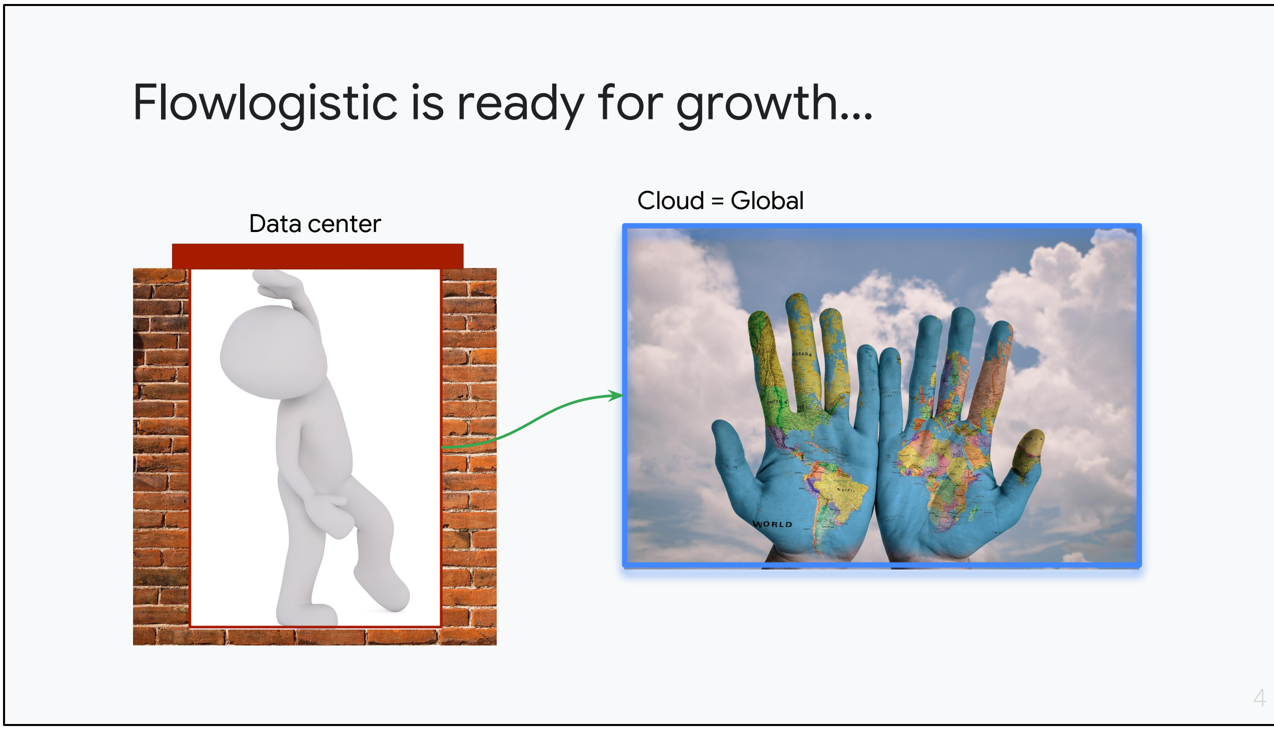
https://pixabay.com/en/arrows-direction-production-planning-1577983/

Key business points

Flowlogistic

Logistics and supply chain provider	Core values	Immediate business goals
Grew from regional trucking company Worldwide rail, truck, aircraft, and ocean shipping Proprietary technology for tracking parcels in real time Unable to deploy because the technology stack (based on Kafka) can't support the volume	Wants to analyze orders and shipments to determine how best to deploy their resources, identify target customers, and market opportunities Use historical data to perform predictive analytics (e.g., when a shipment will be delayed)	Overcome scaling limits of the data center Real-time inventory tracking system that indicates the location of their loads Analytics on orders and shipment logs (with structured, unstructured data) Predictive analytics

Note: The case information does not explicitly describe how information is transmitted from assets back to the data center. We must assume that connectivity is in place for sending tracking information in real time from their resources. If these are tracking devices, it might indicate… using Cloud IoT Core to manage the device connections and Cloud Pub/Sub to buffer and aggregate messages. It seems like this is essential to the solution, overcoming Kafka's design-based bottleneck with Cloud Pub/Sub.



The data center has become a ceiling for growth. The data center can't keep up and it is now a key limiting factor.
Moving to cloud will enable global expansion.

https://pixabay.com/en/movement-stretch-over-head-2566561/

https://pixabay.com/en/hands-world-map-global-earth-600497/
https://pixabay.com/en/brick-wall-red-brick-wall-wall-3364411/

Business requirements

Build a **reliable** and **reproducible** environment with **scaled parity of production**.
Aggregate data in a centralized data lake for analysis.
Use historical data to perform **predictive analytics** on future shipments.
Accurately **track every shipment** worldwide using proprietary technology.
Improve **business agility** and speed of innovation through **rapid provisioning** of new resources.
Analyze and optimize architecture for performance in the cloud.
Migrate fully to the cloud if all other requirements are met.

TIP: Separate business from technical requirements. Look for keywords and phrases that map to or imply data engineering solutions.

Data Engineers should document a set of requirements. Best practice is to separate business requirements from technical requirements.

Common executive motivations:

- Create more revenue opportunities
- Reduce costs to increase profits
- Differentiate from competitors

In the actual data engineer job, conversations tend to use the language of the industry instead of the language of data engineering. Technical leaders will probably discuss *infrastructure* and *architecture* instead of communicating in data engineering terms. You need to consider these statements in the context of the business requirements to identify elements that are important to the data engineering solution. In the requirements listed, keywords and phrases are bolded. These keywords are clues that help the data engineer narrow the possible solutions to a specific or best solution. The same is true for exam questions. The hints are in the case and question. Look for keywords.

The highlighted words are examples of important information that will help drive data engineering recommendations.

Example:

Keywords and phrases like “track every shipment,” “aggregate,” and “analytics” map perfectly to the ETL paradigm: Extract, Transform, and Load.

Phrases like “rapid provisioning” and “business agility” also suggest that speed and latency should be important considerations in the solution.

On the exam, the technical requirements will often result in a couple of equally likely candidate solutions. You should use the business requirements as the "tie breaker" to determine the solution that is best for the business and not just technically feasible.

Technical requirements

Handle **both streaming and batch data**.
Migrate **existing Hadoop workloads**.
Ensure that architecture is **scalable and elastic** to meet the changing demands of the company.
Use **managed services** whenever possible.
Encrypt data in flight and at rest.
Connect a VPN between the production data center and cloud environment.

TIP: Use technical requirements to identify candidate solutions. Look for keywords and phrases that map to or imply data engineering solutions that map directly to GCP products or limit the solution to a group of products.

Keywords and phrases are bolded that point toward the data engineering solution. The technical requirements are the most important indicators and guide for shaping the data engineering solution.

“Managed services” means they want to do as little infrastructure administration as they can. Google Cloud in general, and big data products specifically, are designed with a managed services philosophy. Remember that a managed service may still have IT overhead and reveals the instance or cluster in use. But a serverless service conceals the existence of instances and eliminates that remaining overhead. So when they say "prefers managed services" it might mean that they also "prefer serverless services" more.

“Migrating Hadoop” points to the client's thinking and maturity with data engineering technology. You should expect to encounter systems like HDFS, Pig, Spark, Hive, and Kafka, popular open-source implementations of different workloads. This is a clear indicator of the GCP products to use in the solution—Cloud Dataproc.

The phrase “both streaming and batch” indicates how they want to transform their data streams. You should immediately be thinking that Cloud Dataflow can process both streaming and batch with the same pipeline solution—so it should be a candidate.

“Encrypt” and “Connect a VPN” highlights the client’s bias toward security. This isn't the core of data engineering solutions on GCP, but it is part of the infrastructure and part of the requirements of the job. So it is important to be familiar with networking and security best practices.

Technical evaluation of existing environment

Location/distribution

Existing solution is in a single data center

Databases

- 2 x SQL Server clusters (8 physical servers)
  - User data, inventory, static data
- Cassandra (3)
- Kafka (10)

Application servers

- 60 VMs across 20 physical servers
  - Tomcat for Java
  - Nginx for static content
  - Batch servers

Storage

- Storage appliances
  - iSCSI for VMs
  - SAN for SQL Server storage
  - NAS for image storage, logs, backups

Data processing

- Hadoop/Spark (10)
  - Core data lake
  - Data analysis workloads

Infrastructure

- Miscellaneous (20)
- Jenkins
- Monitoring, bastion hosts, security scanners, billing software

Machine learning and predictive analytics

No existing environment for predictive analytics

7

TIP: These seven items are a great way to organize evaluation of the case/question.

1. Location/distribution (Architecture)
2. Storage
3. Databases
4. Data processing
5. Application servers
6. Infrastructure
7. Machine Learning

Technical watchpoints

Handle both streaming and batch data.
Migrate existing Hadoop workloads.
Ensure that architecture is scalable and elastic to meet the changing demands of the company.
Use managed services whenever possible.
Encrypt data in flight and at rest.
Connect a VPN between the production data center and cloud environment.
GCP doesn't offer managed services for SQL Server or Cassandra; *migrate to GCP alternatives?*
NAS seems to be used for administration (images, logs, backups) (mainly as raw network storage) instead of application files ("filer" -- as a file system).

TIP: Take notice of version issues.

In practice, in the data engineering job, you need to be concerned about versions. For example, it is common for organizations to run old versions of Hadoop and not update until support for components is dropped. Some companies have a "leading edge" philosophy and want to be early adopters of new Hadoop technology. But most companies believe that updates are risky and costly and usually produce little or no added benefit. That raises the issue of whether the current Hadoop applications in use in the data center are compatible with the version of Hadoop (and Hive, and Pig, and Spark, and Python) running in Cloud Dataproc.

Define your solution

Pause and define your own solution, before continuing to view an example solution.

Proposed solution (Part 1)

Networking and connectivity VPC covering multiple regions Cloud VPN (potentially over peering) or Dedicated Interconnect
Applications Lift-and-shift to Compute Engine. Watch out for the local disk requirements. For persistency, use Persistent Disk; but be careful, because it has lower IOPS than local disks.
Optimize and migrate to GCP equivalents/replatform Replace Kafka with Cloud Pub/Sub. Replaced data collection method (not given) with data collected using IoT Core. SQL Server: Depending on use case and size, perhaps replace with Cloud SQL or Cloud Spanner. Cassandra: Consider replacing with Cloud Datastore or Cloud Bigtable.
Hosted applications Compute Engine, Kubernetes Engine, App Engine
Static content Cloud Storage with Content Delivery Network (CDN)

Proposed solution (Part 2)

Batch server workloads.
Use Compute Engine preemptible VMs if the jobs are resilient to failure. Consider Persistent Disks, but be careful of lower IOPS than local disks.
Use App Engine with cron scheduling for repeated workloads/processing.
Consider using Cloud Composer (Apache Airflow) for coordinating data processing workflow.
Consider migrating workflows from Cloud Dataproc to Cloud Dataflow for horizontal scalability to shorten processing time of batch workloads. This would depend on the value of shortening processing time of the particular job. As an alternative, consider adding preemptible VMs to the Cloud Dataproc cluster to avoid migration to the Cloud Dataflow pipeline.
Use Cloud Storage for images. Use Stackdriver Logging for logs.
Backups: Persistent Disk snapshots. PDs can be mounted as read-only.
Data lake: Cloud Storage in place of HDFS. Potential migration to BigQuery for structured data.

TIP: In some cases there is a direct equivalent solution, such as using Cloud Dataproc for Hadoop with HDFS. But there might be a better solution using Cloud Storage instead of HDFS. And for some workloads, migrating from Cloud Dataproc to BigQuery. They key here is to consider the business requirements and whether the client is ready to jump to new technology or would prefer a more gradual path to cloud adoption.

Proposed solution (Part 3)

Hadoop/Spark workloads (Batch and streaming)
Cloud Dataproc Potential migration from Spark SQL to BigQuery
BigQuery streaming insert Dataflow for batch/streaming
Jenkins to trigger pipelines in Spinnaker
Stackdriver for monitoring, logging; BigQuery for logging
Kubernetes Engine/Cloud Engine/App Engine for billing software
Cloud Datalab
Cloud ML Engine and ML APIs

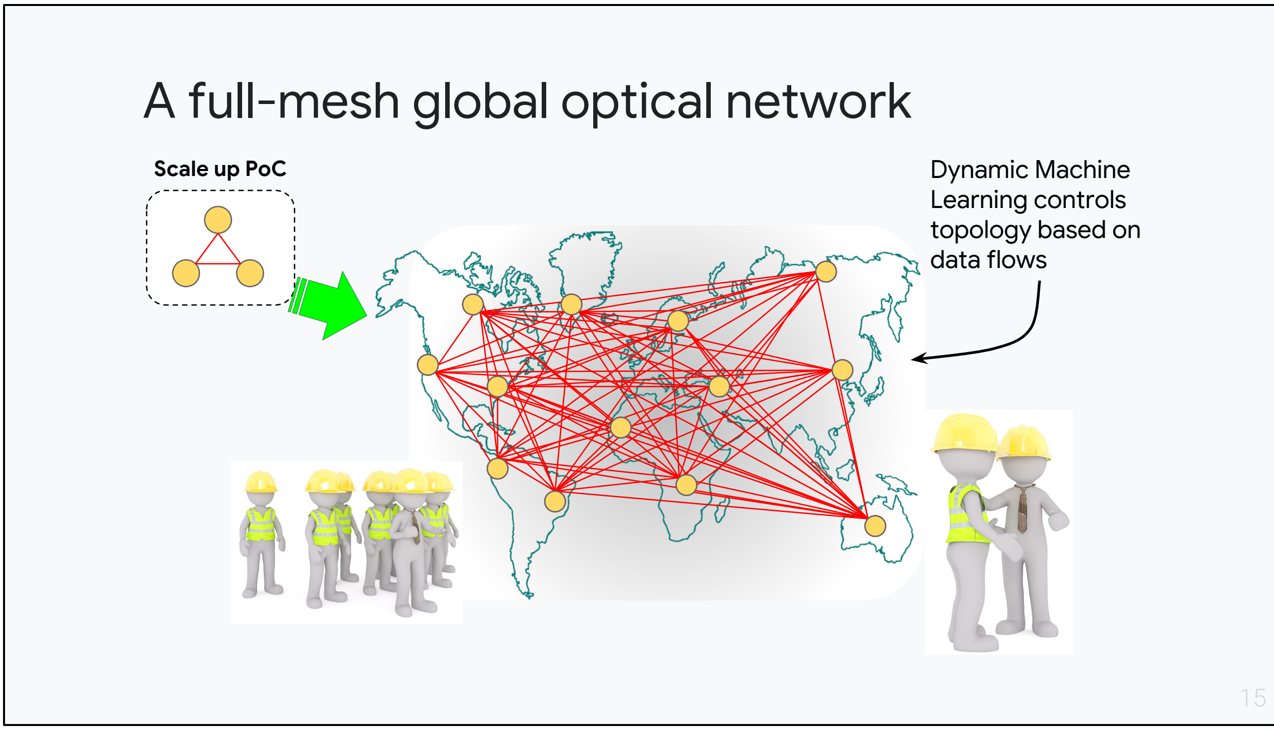
The case is no longer provided by the Certification organization, but is now offered in this course for training purposes.

https://pixabay.com/en/conduit-pipes-coils-rolls-colours-166802/

Key business points

MJTelco

Startup	Core values	Immediate business goals
Optical networking startup Inexpensive innovative proprietary optical hardware Provides backbone networks to underserved markets Proof of concept already successful	Continuously optimize topologies Over-deploy to mitigate regional politics	Distributed data infrastructure that drives real-time analysis and incorporates machine learning **Scale** PoC to 50,000 installations **Harden** (secure) solution **Improve ML** used for defining topology by refining machine learning cycles

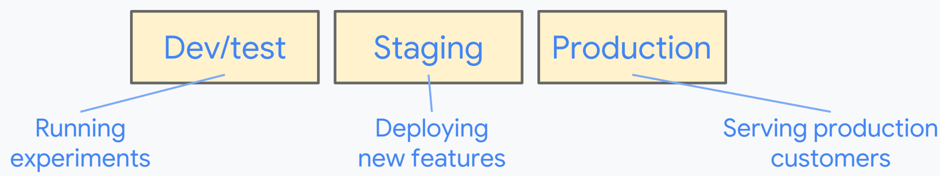


Solution concept. Optical networks generally can't route data the way electronic networks can. The optical components necessary to switch data are only recently being developed and are still experimental. Consequently, optical networks are formed of point-to-point links. Data is forwarded either by being optically split and enhanced (using an erbium-based optical amplifier) or by being converted to electrical signals and then regenerated to optical signals for the next hip in the link. Converting a signal from light to electrons and back is a slow process relative to optical speeds. So understanding the delay involved and when the data is needed is important to deciding how to transmit the data flows. In a full-mesh optical network, any data flow could be sent directly from one point to another. However, for capacity management, it might make sense to aggregate data from point "A" and send it to point "B" and then split or re-generate the signals for forwarded data from "B" to "C," instead of sending data separately from "A" to "B" and "A" to "C." Reconfiguring the topology of such a network in advance of the planned data flows is key to managing such a design. Machine learning is used to predict optimal models for the network. As the netwok grows in utilization, the ML models should get better, thus making the network more efficient.

https://pixabay.com/en/construction-workers-black-workers-2606310/
https://pixabay.com/en/construction-manager-2606301/
https://pixabay.com/en/map-silhouette-map-contour-map-961700/

Business requirements

Scale up production environment with **minimal** cost, instantiating resources when and **where needed** in an unpredictable, distributed telecommunication user community.
Ensure **security** of their proprietary data to protect their machine learning and analysis.
Provide **reliable** and **timely access to data** for **analysis by distributed research workers**.
Maintain **isolated environments** that support **rapid iteration of their ML models** without affecting their customers.

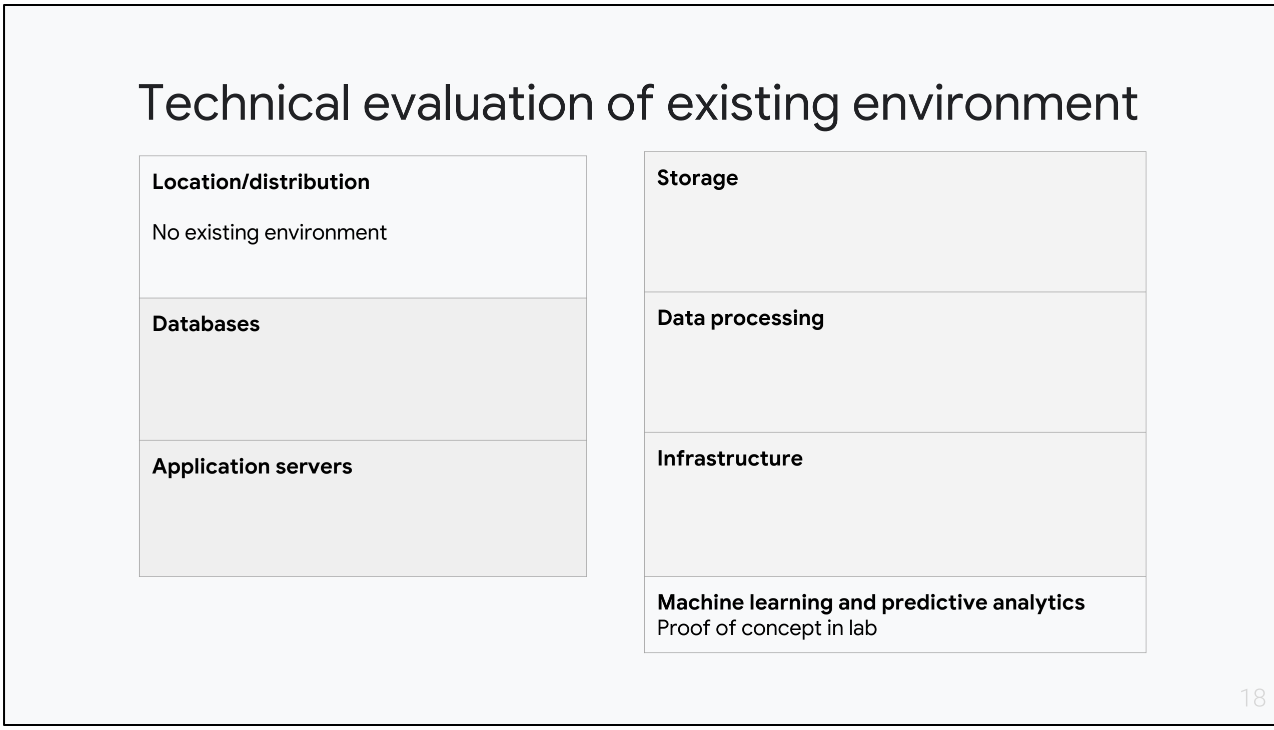


TIP: Separate business from technical requirements. Look for keywords and phrases that map to or imply data engineering solutions.

CEO: "We already have a cost and reliability advantage. We need help meeting capacity (scale) and data security goals."
CFO: "We need our analysts focused on product improvements in machine learning, not solving operational problems in our data pipeline."
CTO: "Scaling and security. Data Analysts innovate for customers. So we need rapid and iterative dev/test."

Technical requirements

Ensure **secure** and efficient **transport** and **storage** of telemetry data.
Rapidly scale instances to support between 10k and 100k data providers with multiple flows each.
Store and analyze 2 years of data at 100 million records/day.
Support **rapid iteration of monitoring infrastructure** focused on awareness of **data pipeline problems** both in telemetry flows and in production learning cycles.



TIP: This case does not give a lot of details about the existing environment or the application itself. All we know is that it is a machine learning application that relies on big data processing, that it has to scale and be secure, and that it involves ML analytics. Therefore, confine your solution to those elements and make reasonable assumptions.

1. Location/distribution (Architecture)
2. Storage
3. Databases
4. Data processing
5. Application servers
6. Infrastructure
7. Machine learning

Technical watchpoints

What are the network and log transfer capabilities of this "inexpensive hardware"?
Should any ETL be performed on the data?
Are any application servers necessary?
Multiple zones are needed for reliability, multiple regions for global expansion.
Is the data structured? Is it relational? Is high throughput required?

Desire for a modular solution that can be "iterated" for new markets, locations, clients.

Define your solution

Pause and define your own solution before continuing to view an example solution.

Proposed solution (Part 1)

Cloud IoT Core to manage security and firmware provisioning of devices and to pipe logs from devices to Google Cloud Platform data processing solutions.	Either approach uses Google networking. A Global IP can be used to access the closest geographic region resources (load balancing); it can then forward the traffic to the correct internal solution as needed.
Or, SFTP (Secure File Transfer) to a storage bucket.	
If application servers are required, App Engine or Kubernetes Engine might reduce IT overhead and provide for more rapid and scalable development and testing.	
Use persistent disks on Cloud Dataproc clusters if persistence is required, or use Cloud Dataproc with Cloud Storage.	
Data lake: Use BigQuery if data is structured. If data is unstructured use Cloud Storage in a multi-regional bucket.	
Hadoop/Spark workloads: Host several different Cloud Dataproc clusters in different regions, all with access to the data lake. Consider using Cloud Bigtable in place of HBASE for high speed access.	

A data warehouse is primarily business transactional and relational data from business logic systems. A data lake is primarily unstructured data from logs, click-streams, and other kinds of dynamic or streaming data such as IoT.

Proposed solution (Part 2)

Use Stackdriver to understand the performance and health of your Cloud Dataproc clusters and examine HDFS, YARN, and Cloud Dataproc job and operation metrics.
Use Stackdriver to monitor Cloud Dataflow if Cloud Dataflow is used.
Cloud ML Engine ML APIs