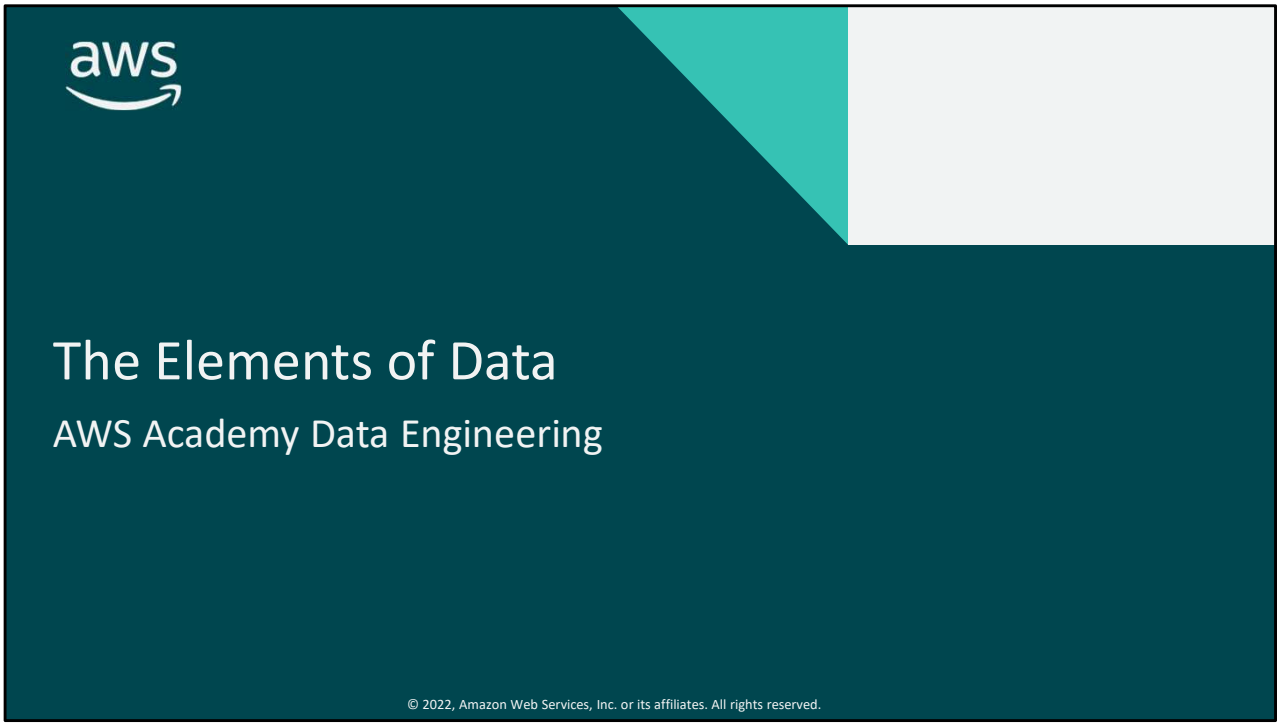
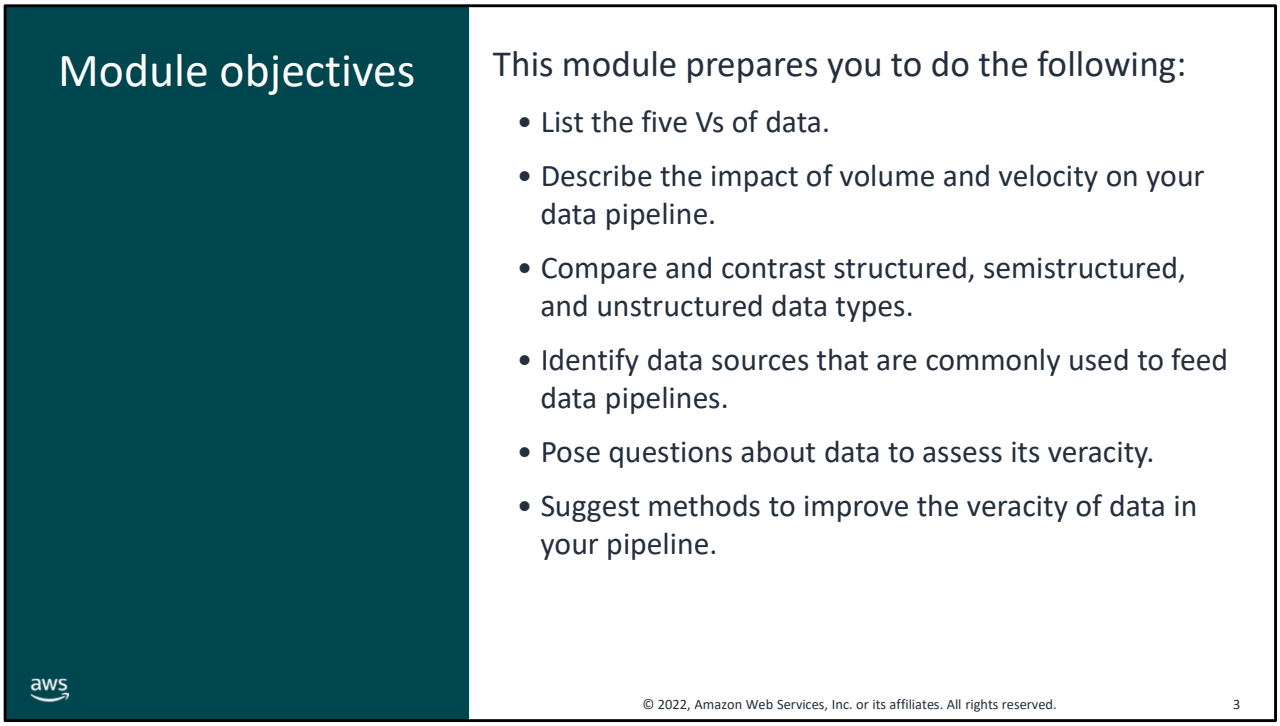


|Slide number 1
|Instructor notes
|
|Student notes
Welcome to the Elements of Data module. This module introduces elements of the data that you need to be aware of to design a data pipeline for decision-making.

|Slide number 2
|Instructor notes
|
|Student notes
This introduction section describes the content of this module.



|Slide number 3
|Instructor notes
|This module is intended to introduce concepts that will be revisited in following modules and iterated on in greater depth. Key points to reinforce are the need to design each pipeline to match the data volume and velocity that are expected as well as the characteristics of the individual data elements that you intend to act on.
|
|Student notes
The module introduces a basic vocabulary to think about the data sources that will feed your pipeline. You will learn about the five Vs of data and how each of them influences the choices that you will make in your data pipeline.

Module overview

Presentation sections

- The five Vs of data – volume, velocity, variety, veracity, and value
- Volume and velocity
- Variety – data types
- Variety – data sources
- Veracity and value
- Activities to improve veracity and value

Activity

- Planning Your Pipeline

Knowledge checks

- Online knowledge check
- Sample exam question

|Slide number 4
|Instructor notes
|
|Student notes
The objectives of this module are presented across multiple sections.

You will also complete a hands-on activity that asks you to consider data types and sources for an example use case.

The module wraps up with a sample exam question and an online knowledge check that covers the material presented.

Common data pipeline questions

Data engineer

- Does the organization have the data that addresses the need? Where is the data stored and in what format?
- Will I need to combine data from multiple sources?
- What is the type and quality of data? What will be the source of truth?
- What are the security requirements for this data? Who needs access to it and in what state?
- What types of mechanisms are needed to transfer the data from its locations into the pipeline?
- How much data is there, and how frequently is it updated or processed?
- How important is speed when data is requested?

- What can the data tell me?
- How will I evaluate the results?
- What kind of visualization do I need?
- Which formats and tools are the analysts familiar with?
- Do I need a big data framework?
- Which type of AI/ML models fit?
- What is the simplest way to implement AI/ML?

Data scientist

aws

|Slide number 5

|Instructor notes

|If there is a time lapse between modules 2 and 3, you might want to review this slide as a prompt. If delivering back to back, you might skip it.

|

|Student notes

The questions on this slide were introduced in the Data-Driven Organizations module. The questions are directly relevant to the data characteristics that are discussed in this module. Answering these questions means understanding the five Vs of data.

|Slide number 6
|Instructor notes
|
|Student notes
This section identifies the five Vs of data, which map to the questions that a data engineer must answer to design a good data infrastructure.

Data characteristics that drive infrastructure decisions

Volume

How big is the dataset? How much new data is generated?

Velocity

How frequently is new data generated and ingested?

Variety

What types and formats? How many different sources does the data come from?

Veracity

How accurate, precise, and trusted is the data?

Value

What insights can be pulled from the data?

aws

|Slide number 7
|Instructor notes
|The goal of this section is to highlight each characteristic, how they intertwine, and the trade-offs involved in making design decisions. Generally it's value first, and then volume/velocity variety/veracity as pairs.
|
|Student notes
The five Vs are volume, velocity, variety, veracity and value.

Consider volume and velocity together because you will make infrastructure decisions about how to collect, store, and process data based on the combination of how much data you need to ingest and how quickly you will ingest it.

Variety and veracity both relate to the data itself—what type of data is it and what's the quality of it. Data engineers and data scientists will transform and organize the data based on its variety and veracity to make it useful for analysis.

Value is about ensuring that you are getting the most out of the data that you have collected. Value is also about ensuring that there is business value in the outputs

from all that collecting, storing, and processing.

Strategies that support getting the best value from data

- Confirm available data meets need.
- Evaluate feasibility of acquiring data.
- Match pipeline design to data.
- Balance throughput and cost.
- Let users focus on business.
- Catalog data and metadata.
- Implement governance.

|Slide number 8
|Instructor notes
|
|Student notes

To work backwards from the business need, let's look at value first. The value that you want from your data will drive all of the pipeline decisions that you make. The data strategies that were highlighted in the Data-Driven Organizations module are reflected in these strategies.

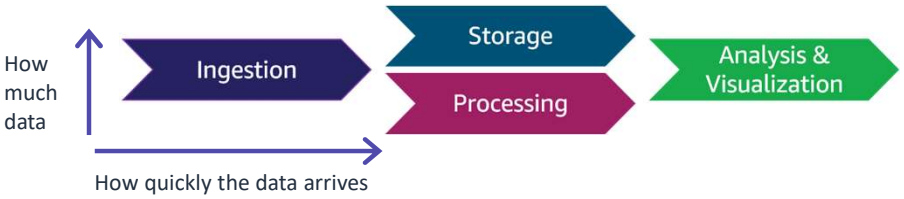
First, decide whether it appears that you can derive the desired outcome from the data sources that you have. Determine if you need to find, buy, or generate other datasets to meet the need and whether it will be feasible and cost-effective to get the data.

Next, match your infrastructure decisions to the volume, velocity, and variety of data to be collected and stored, and find the best balance of throughput and cost for each use case. Where is it worth spending more to move faster or acquire more data for decision-making?

After you decide to invest in the data, maximize the ability of your users to draw out insights. This means reducing the time that you spend focusing on operational or administrative tasks rather than directly focusing on business value. This also means making it possible for users to discover all the data that you already have that might be relevant to a new business problem. It's also critical that you put governance in place to maintain the trustworthiness and accuracy of your data.

|Slide number 9
|Instructor notes
|
|Student notes
This section discusses how the volume and velocity of data can shape your pipeline.

Scaling your pipeline for volume and velocity of data



- The amount of data plus the pace of data drive design choices.
- Volume and velocity impact all layers of the pipeline.
- Evaluate each pipeline layer for its own requirements.
- Balance costs for throughput and storage against the required time to answer and the accuracy of the answer.

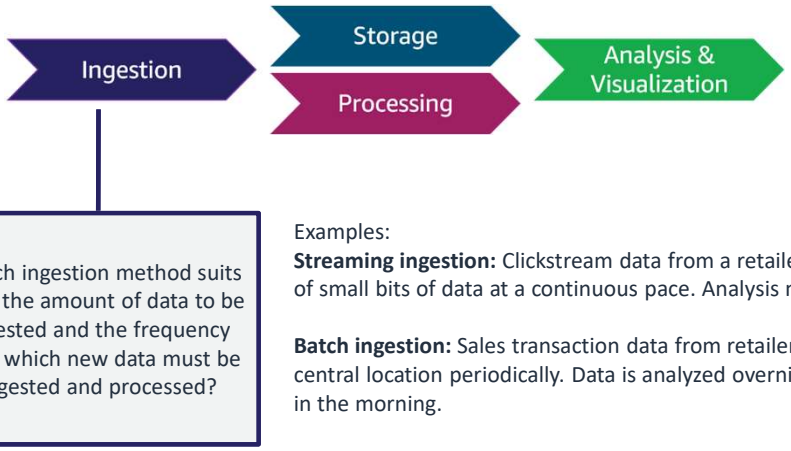
|Slide number 10
|Instructor notes
|
|Student notes
Volume and velocity requirements drive infrastructure decisions across all pipeline layers. Although they are different measures, you need to look at them together to determine how your pipeline needs to scale. You need to scale to handle ingestion and storage of the amount of data at the pace of its arrival. You also need to consider how long data should be stored to balance cost and availability.

From the processing standpoint, you need to understand how much data must be processed and analyzed to address a singular business problem. You also need to know how quickly the data must be processed after it arrives in the pipeline.

In terms of visualizing data, you need to understand how much data consumers need to access at one time and how frequently new data must be incorporated.

The next few slides give examples of the types of decisions you will need to think about that are related to volume and velocity.

Ingestion decisions related to volume and velocity



Examples:
Streaming ingestion: Clickstream data from a retailer's website sends a large volume of small bits of data at a continuous pace. Analysis must be performed immediately.

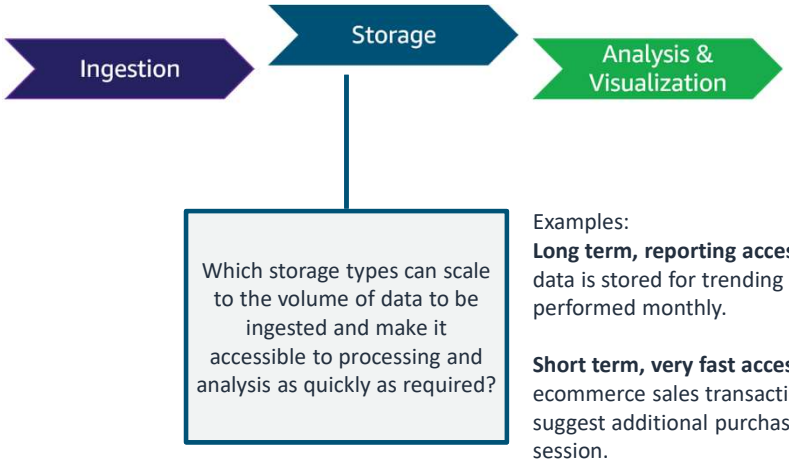
Batch ingestion: Sales transaction data from retailers across the world is sent to a central location periodically. Data is analyzed overnight with reports sent to branches in the morning.

aws

|Slide number 11
|Instructor notes
|Successive modules will present more details on ingestion methods and ingestion types, such as batch, transactional, and streaming, and will present the types of services that suit each. The goal of this module is to start students thinking about the factors that will drive those decisions.
|
|Student notes
To choose ingestion methods for your pipeline, you need to understand the volume of data it's expected to handle at a given interval. For example, you might need to ingest a continual stream of large volumes of small records that need to be processed as quickly as possible. Or you might need to ingest larger individual records that arrive periodically and are stored to process as a batch of records.

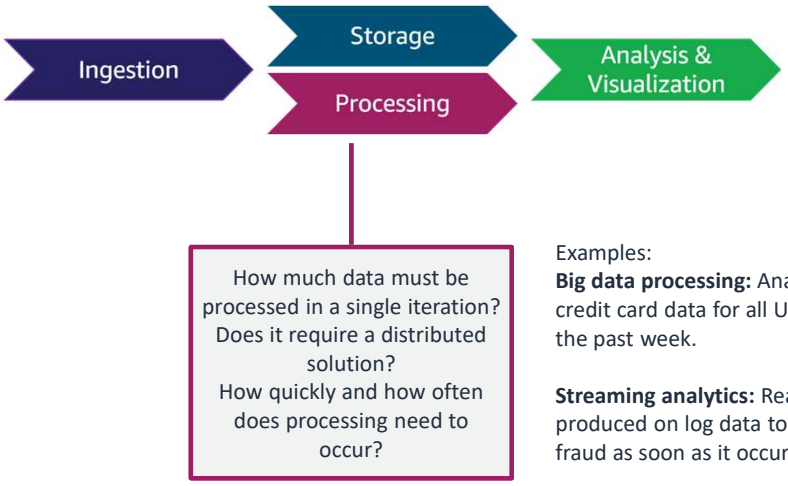
Both of these examples include high volumes, but the velocity of their arrival and the speed with which they must be processed impact your pipeline design.

Storage decisions related to volume and velocity



|Slide number 12
|Instructor notes
|Successive modules will present more details on storage decisions and how purpose-built stores are used to match data volume, velocity, and type. The goal of this module is to start students thinking about the factors that will drive those decisions.
|
|Student notes
Your storage decisions will be closely tied to ingestion methods and the volume and velocity of incoming data. You will also need to consider how the data will be accessed. For example, will it reside in storage for periodic access over a long period of time, or will it be stored only briefly before it loses value?

Processing decisions related to volume and velocity



|Slide number 13
|Instructor notes
|Successive modules will present more details on processing and go deeper into big data and machine learning (ML) processing. The goal of this module is to start students thinking about the factors that will drive decisions about the type of processing framework that they will need.
|
|Student notes
The distinct needs for processing high volumes of data—that is, *big data*—have been the driver behind big data frameworks (for example, Apache Hadoop and Spark). These frameworks are designed to process really big chunks of data very quickly by using a distributed system. The volume, frequency, and type of processing that you need to do to find business insights might lead you to use a big data framework.

Analysis/visualization decisions related to volume and velocity

Examples:
Historical analysis: A year's worth of sales data is visualized, and users can drill down by region and salesperson.

Streaming Internet of Things (IoT) data: The real-time error rates of sensors in a plant are visualized.

How much data must be used in a single visualization? Do users need to drill down into details?
How quickly do consumers need to see and act on the data?

|Slide number 14
|Instructor notes
|Successive modules will present more details on analysis and visualization and go deeper into different tools based on the type of analysis to be done. The goal of this module is to start students thinking about the factors that will drive decisions about the type of tools that they will need.
|
|Student notes
Analysis and visualization tools must match the volume and velocity of the data being ingested and processed. Visualizations that aggregate data must be able to scale to match the volume of data that is being analyzed. Elements that must be reviewed in real time need to be made available in tools with low latency.

Key takeaways: Volume and velocity

- Volume is about how much data you need to process.
- Velocity is about how quickly data enters and moves through your pipeline.
- Volume and velocity together drive the expected throughput and scaling requirements of your pipeline.
- You should evaluate the volume and velocity requirements for each layer in your pipeline.
- Balance costs and throughput requirements.

|Slide number 15
|Instructor notes
|
|Student notes

Here are a few key points to summarize this section.

- Volume is about how much data, and velocity is about the pace of data. Together, they drive the scaling requirements for your pipeline.
- When you build your pipeline, consider volume and velocity at each layer. For example, some data that arrives at high velocity doesn't need to be processed immediately and can be stored and processed at a slower pace.
- Your choices should balance costs for throughput and storage against the required time to answer and accuracy of the answer.

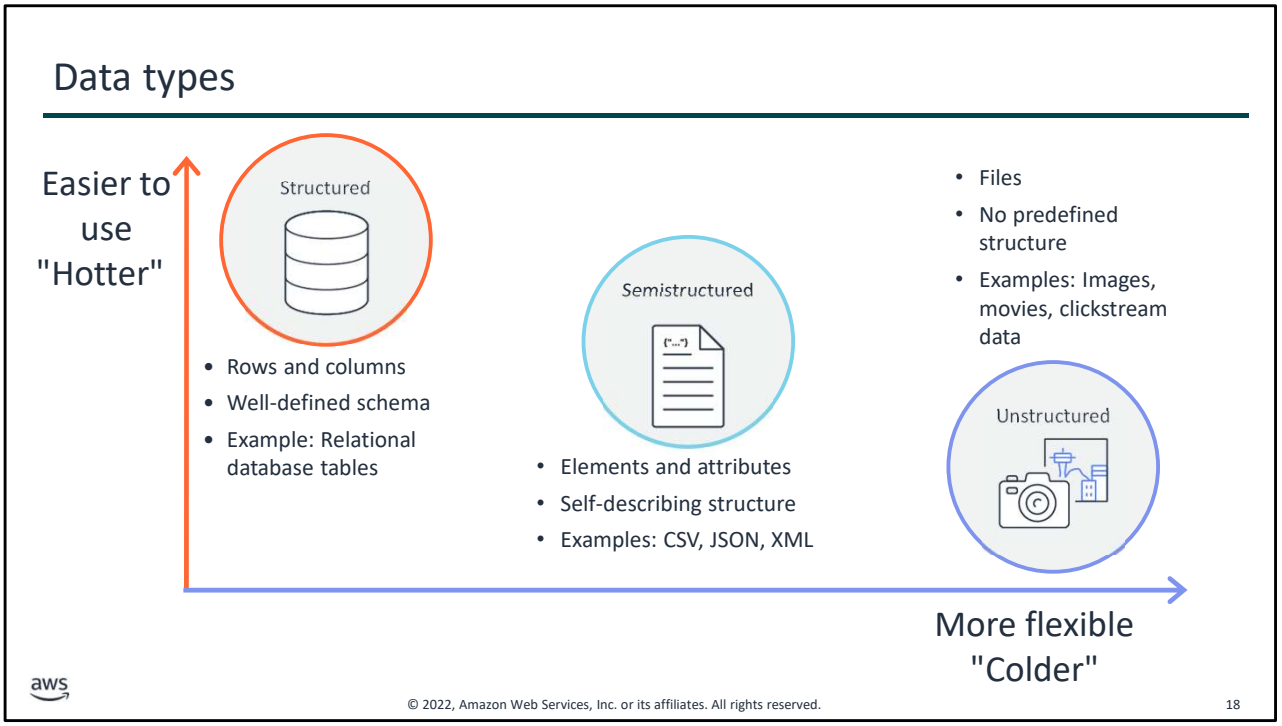
|Slide number 16
|Instructor notes
|
|Student notes
This section discusses the variety of data types that you will need to use in your pipeline.

Designing your pipeline to support data variety

When deciding on the data that you will use, the variety of both data types and data sources will influence pipeline design on each layer of your pipeline.

- Different data types will lend themselves to certain types of processing and analysis.
- Different data source types might require different amounts of discovery and transformation work.
- The data source type will also drive the type and scope of the ingestion layer.

|Slide number 17
|Instructor notes
|This module looks broadly at considerations for three data types and three data source types. Successive modules will return to these topics as part of specific use cases or patterns. The goal of this section is for students to recognize the main distinctions and types of things that a data engineer needs to think about in regard to how data will pass through the infrastructure.
|
|Student notes
In this section, you will learn about a few general data types. This section will be followed by one about data source types. These sections will help you start to think about how these characteristics will influence the decisions that you make for your data pipeline.

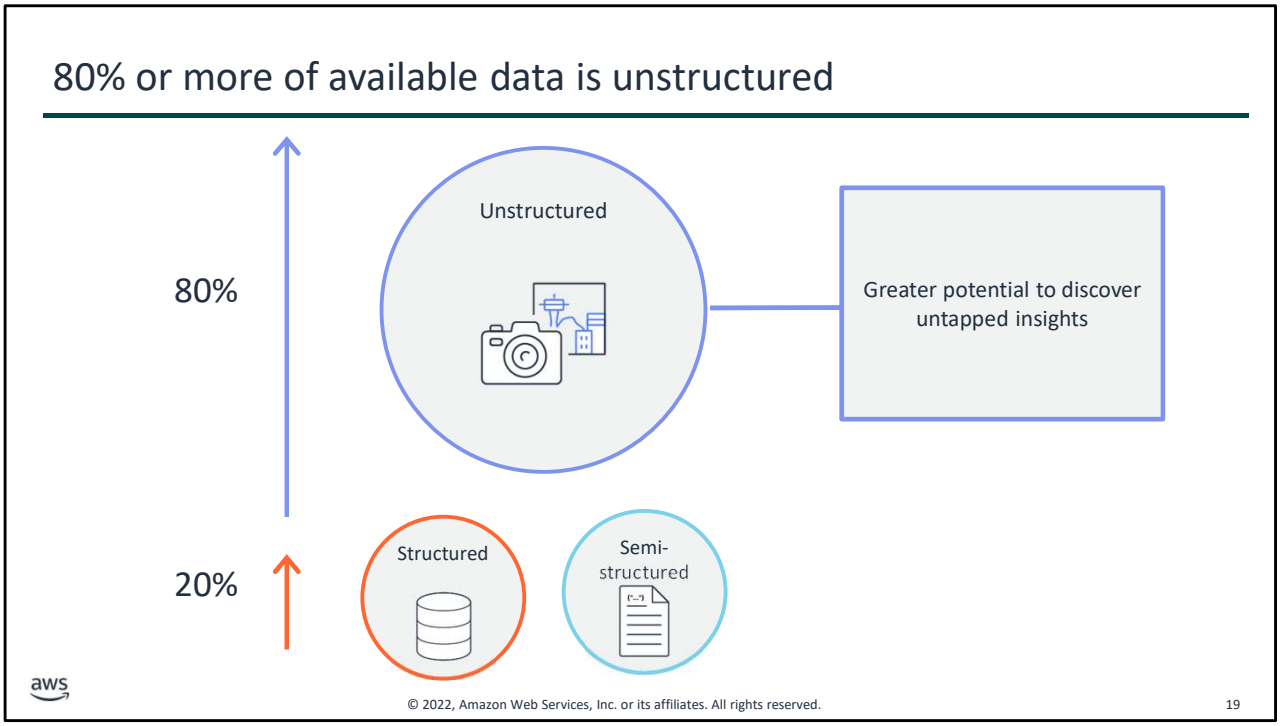


|Slide number 18
|Instructor notes
|
|Student notes
You can separate data into three general types: structured, semistructured, and unstructured.

Structured data is probably what you think about when you hear the word *data*. Structured data is stored in a tabular format, such as records in a traditional relational database. Data is organized based on a data model, which defines and standardizes data elements and their relation to one another. This makes structured data easy to query but not very flexible. Structured data is referred to as being hot—meaning that it's immediately ready to be analyzed.

Semistructured data has a self-describing structure and recognizable elements, but it doesn't have the rigid schema constraints of structured data. For example, a JSON or XML file is semistructured. You can think of semistructured data as lukewarm—some data will be ready for analysis immediately, but it's more likely that data needs to be cleaned or preprocessed before you can analyze it.

Unstructured data doesn't have a predefined structure. This makes it very flexible but more difficult to query. Unstructured data is considered cold and takes the most work to analyze.



|Slide number 19
|Instructor notes
|
|Student notes
Historically, much of the available data and most of the data used in decision-making has been structured or semistructured data (for example, analysts writing SQL queries against a relational database).

But the explosion of data that has been happening is unstructured, such as clickstream data, social media posts, video and image files, and sensor data. Therefore, the focus for a lot of new data infrastructure is on pulling the value out of unstructured data, but as you just learned, getting at that value is challenging. That's where organizations will want to innovate with artificial intelligence and machine learning (AI/ML) to tap into new insights.

Example use cases for data types

Structured	Semistructured	Unstructured
Query a relational database to report on customer service tickets that were submitted in in a specific period.	Extract and analyze customer comments from an online chat application that saves conversations as JSON.	Perform sentiment analysis on customer service emails.

|Slide number 20
|Instructor notes
|This is a good opportunity to prompt students for additional examples of each data type and their use cases. You could also discuss the type of elements or attributes that you might pull out into analysis.
|
|Student notes
This slide lists a few simple use cases for the data types, based on customer service data. A structured example is querying a relational database to report on customer service tickets. A semistructured example is analyzing customer comments that are stored in JSON files. The unstructured example is to use an AI tool to perform sentiment analysis on customer service emails.

Key takeaways: Variety – data types

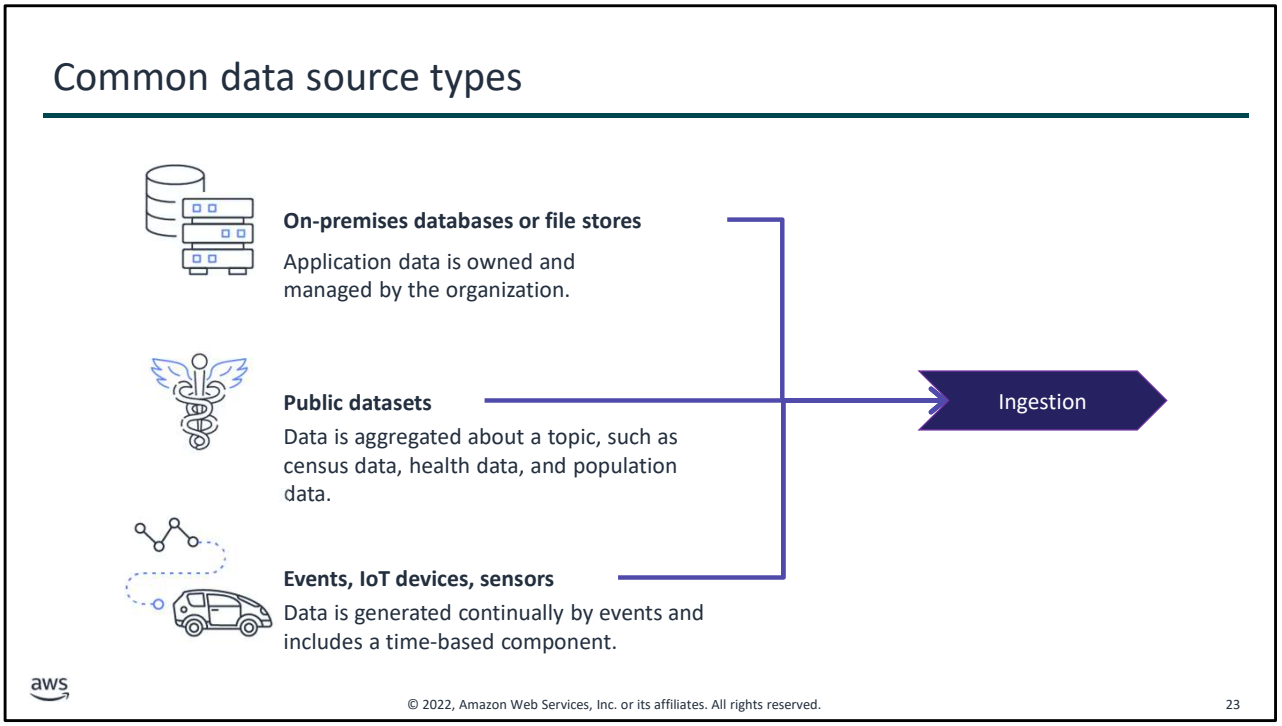
- General data types include structured, semistructured, and unstructured.
- Structured data is the easiest to query but the least flexible.
- Unstructured data is the hardest to query but the most flexible.
- Most of the growth of available data is unstructured.

|Slide number 21
|Instructor notes
|
|Student notes

Here are a few key points to summarize this section.

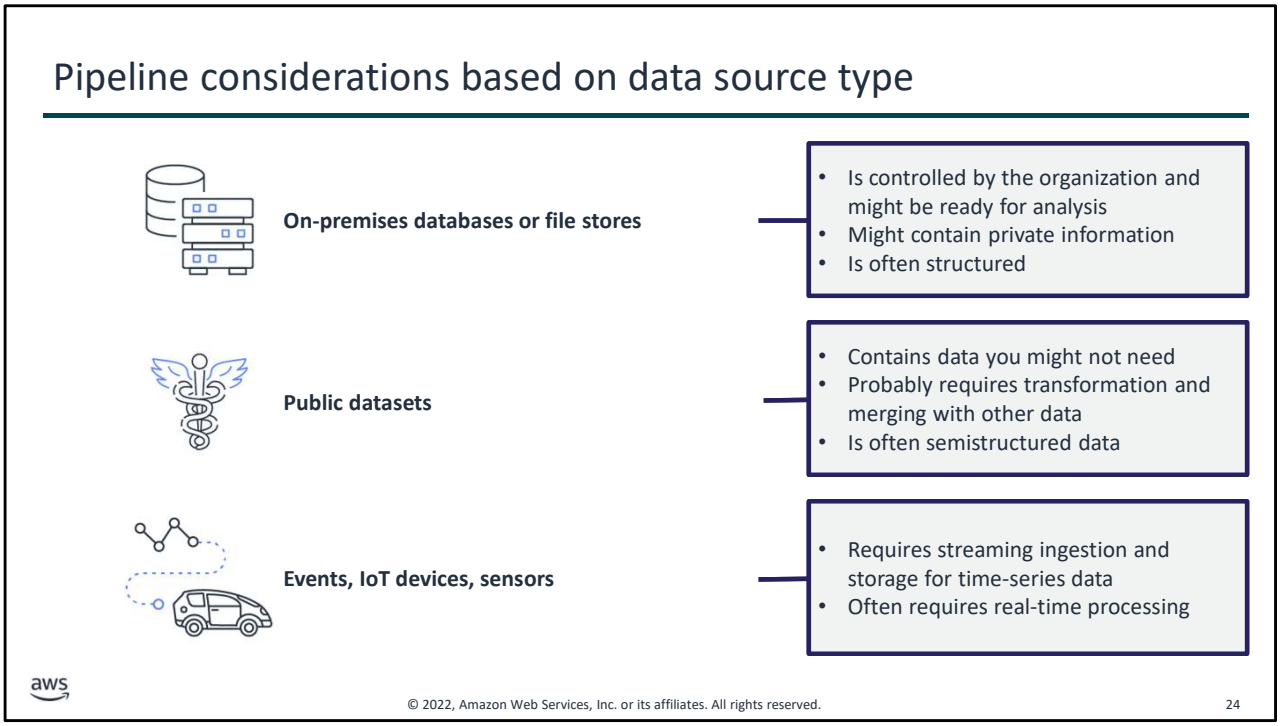
- The three general types of data are structured, semistructured and unstructured.
- Structured data, such as a relational database, is easy to query and process but not very flexible.
- Unstructured data on the other hand is very flexible but more difficult to work with. Most of the data growth in recent years is of the unstructured type.

|Slide number 22
|Instructor notes
|
|Student notes
This section discusses the variety of data sources that you might use to fill your pipeline.



|Slide number 23
|Instructor notes
|
|Student notes
Now that you have looked at how data might be structured, you want to consider the characteristics of the data source itself. Your pipeline might need to combine data from many different sources and pull them together as a new data source to be analyzed.

Three common types of data sources include your organization's own databases or file servers; public datasets about a topic of interest; and data generated by events, Internet of Things (IoT) devices, and other sensors.



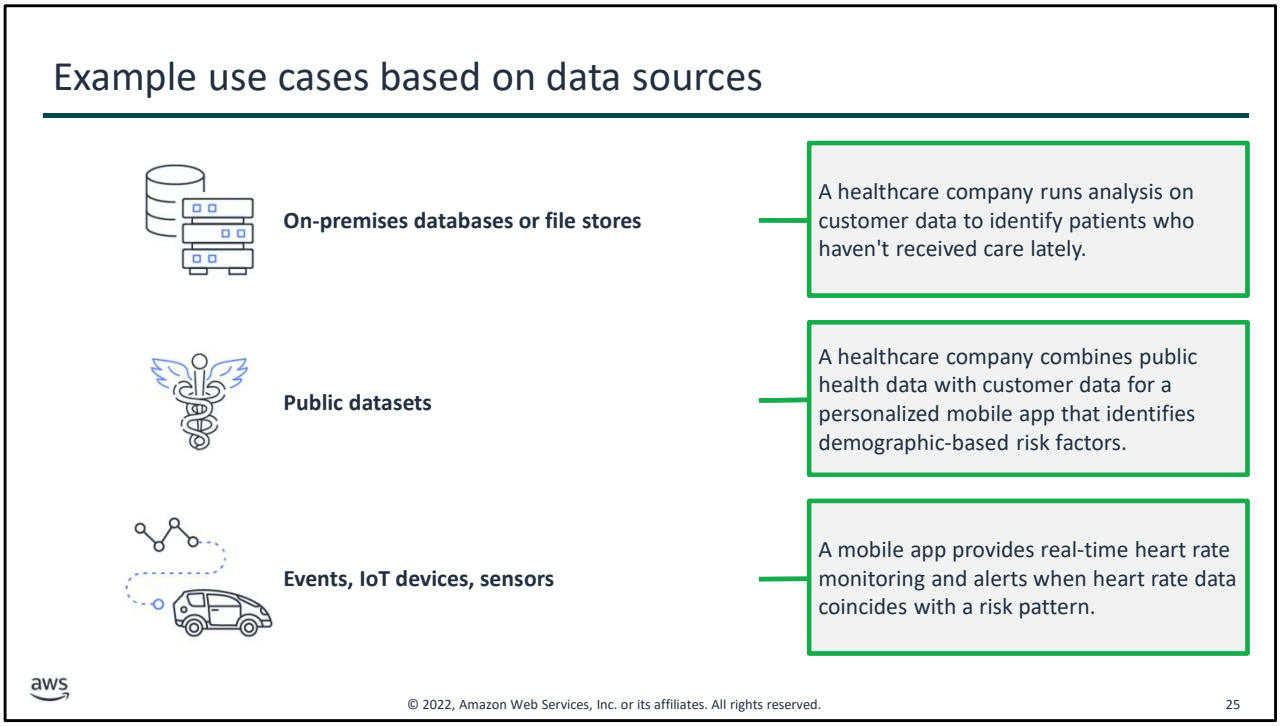
|Slide number 24
|Instructor notes
|
|Student notes
You will need to perform discovery activities that are specific to any data source that you plan to use, but here are some general considerations by source type.

Data sources that are under your organization's control might be structured sources that are ready for analysis in their raw state without any additional massaging of the data. Be cognizant of any personally identifiable information that might be part of the data store. Services are available to load data from on-premises resources into your cloud pipeline, but this might be an opportunity to migrate an on-premises data store to the cloud.

Public datasets are often in a semistructured format and probably contain elements that you don't need and therefore want to discard rather than store. You will also likely need to transform the data to merge it with your own datasets.

Event, IoT, and other time-series data requires your pipeline to handle streaming

ingestion, and you will want to choose a data store that works well for time-series data. These types of data sources lend themselves to real-time processing so that automated actions can be taken based on some event or threshold value.



|Slide number 25
|Instructor notes
|This is a good opportunity to prompt students about other examples they have for each data source type. You could also discuss the type of elements or attributes that might be pulled from each type of data source to create a combined data source.
|
|Student notes
This slide lists a few simple use cases by data source type, using the example of healthcare data. The examples highlight how the data might be combined to help customers.

Data from the on-premises database could be queried to find patients who are overdue for a visit.

By using public healthcare datasets, the company might combine public health data that identifies demographic risk factors for heart attacks with customer data to personalize information for a patient who shares the risk factors. If patients had a mobile app that sent real-time heart monitoring data at some interval, the data could be evaluated for anomalies that coincide with known risks. Then, the patient or their

doctor could be alerted.

You can see in these examples how pulling data from many sources can provide different levels of personalization and prediction.

The challenges of variety

- The way the data has been formatted and stored might impact your ability to analyze it.
- Ingestion and processing methods can become complex when you must combine different data types and sources.
- Data veracity can be more difficult to maintain when multiple data types and sources must be merged.

|Slide number 26
|Instructor notes
|
|Student notes
Although there are benefits to being able to pull from different data types and sources, there are also challenges. The available data type might limit the types of analysis that you can perform. Pulling sources together complicates ingestion and processing and might make it more difficult to manage the veracity of your data.

Key takeaways: Variety – data sources

- Data source types include organizational stores, public datasets, and time-series data.
- Combining datasets can enrich analysis but can also complicate processing.

|Slide number 27
|Instructor notes
|
|Student notes

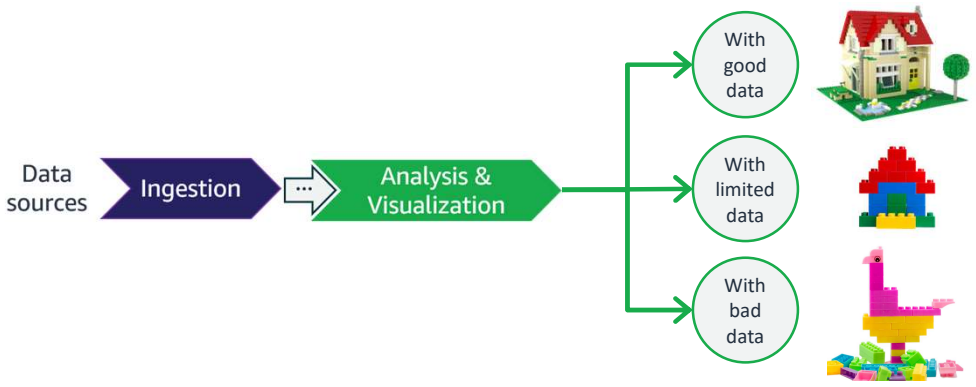
Here are a couple of key points to summarize this section.

- Common data sources include an organization's own databases; public datasets; and time-series data, such as events, IoT data, and sensor data.
- Combining these datasets in your pipeline can enrich your analysis and the decisions that you can support. But, complexity also increases because you must manage the differences in structure and content between each data source.

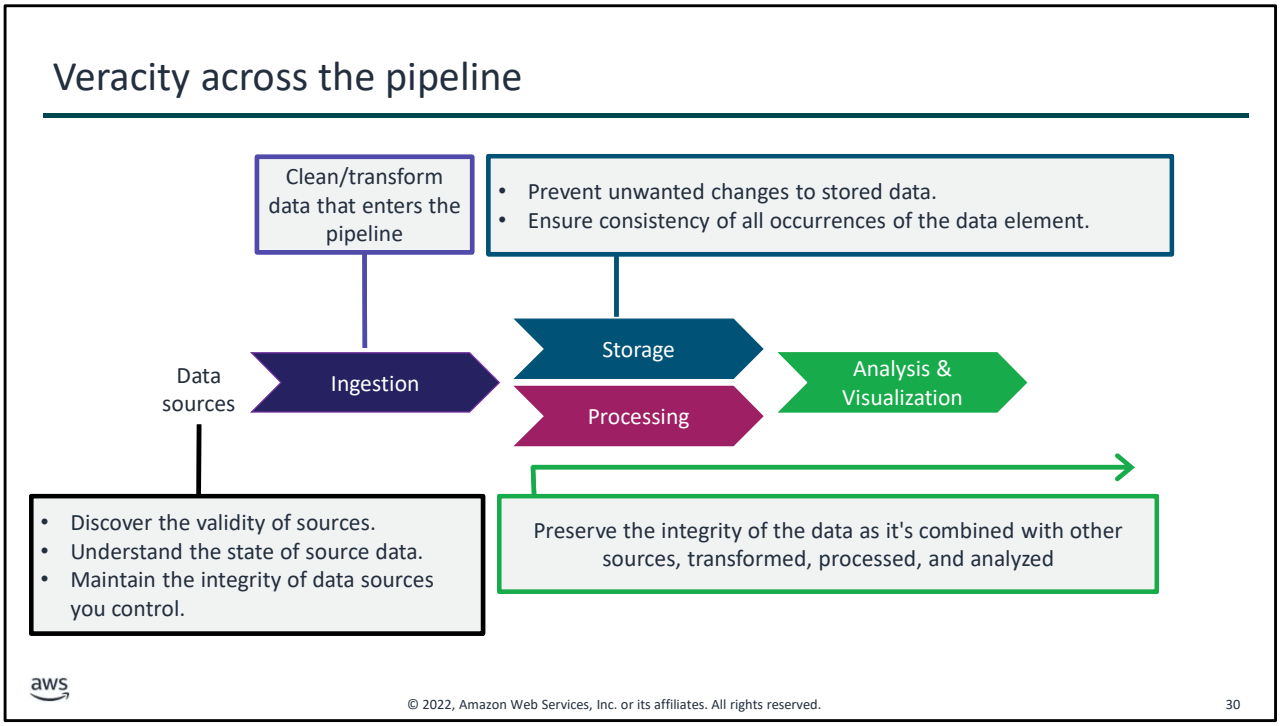
|Slide number 28
|Instructor notes
|
|Student notes
This section discusses considerations for data veracity and its impact on value.

Value rests on veracity

Making a data-driven decision with bad data is worse than making a decision without data.



|Slide number 29
|Instructor notes
|This is a light-hearted analogy, but it might be worth calling out the distinctions it aims to show. With good data, you get a well defined model with lots of details realized. If you have less data, but it is still good quality data, you can still derive the concept of a house from the data, just with much less clarity. If you attempt to process the model with bad data, your outcome is something completely different (in this case a chicken).
|
|Student notes
It might seem obvious, but it's important to remember that the impact of bad data can be worse than working without much data. When you work with limited data, decision makers are aware of the limitations and can manage risk accordingly. If consumers trust bad data to drive decision-making, the results could be dramatically different.



|Slide number 30
|Instructor notes
|
|Student notes
Data veracity is contingent on the integrity of the data.

You might need to determine the integrity of the data source and adjust any areas where that source might lack integrity. Data changes over time. As it is transferred from one process to another and through one system and another, there are opportunities for the integrity of the data to be negatively impacted. You must ensure that you maintain a high level of certainty that the data you are analyzing is trustworthy.

Understanding the full lifecycle of your data and knowing how to protect it effectively will greatly strengthen the integrity of your data.

As part of data discovery, you need to assess how good the data is to start with. As part of ingestion, clean and transform data to make it usable. After you ingest the cleaned data, protect it where it's stored and in each step of its processing.

Examples of data issues that decrease veracity

Discover	Clean/transform	Prevent
Examples • Dated information • Missing data • Lack of lineage • Ambiguity • Statistical bias	Examples • Duplicates • Abnormalities • Source differences	Examples • Software bugs • Tampering • Human error

aws

|Slide number 31
|Instructor notes
|
|Student notes

The first step to understand the trustworthiness of your data is to get more familiar with the dataset's characteristics. For example, how old is the information? Does it seem like the dataset is missing information? What do you know about its lineage? That is, do you know what changes or transactions brought it to its current state? If you have a question about the data, can you go back and understand how its current value was set? A lack of lineage also makes it more difficult to address potential issues such as ambiguity (Does this value refer to Paris, Texas or Paris, France?) and statistical bias (the dataset isn't representative of its population). The level of discovery that you do will depend on what you know about the data source. You might actually perform processing or analysis on the data to help determine its veracity.

After you have a better understanding of the source data, you can take steps to improve its current state and prevent it from becoming untrustworthy. As part of ingesting the data, identify and remove duplicates, find outliers in the data, and

resolve differences in how different sources represent the data.

To protect the data's integrity, you want to prevent software bugs from reintroducing what you removed during cleaning. You also need to prevent unwanted changes—whether they are an intentional disruption, such as a hack, or a mistake made by someone who has access to update the data. Even where you cannot prevent mistakes, introduce appropriate data audits so that you can maintain lineage of the data after it's in your pipeline.

Key takeaways: Veracity and value

- Veracity is about being able to trust the data that you are basing your analysis on.
- Value rests on veracity because without good data you could make bad decisions.
- You need to evaluate the veracity of each data source, clean and transform the data for your needs, and prevent unwanted changes to the data.

|Slide number 32
|Instructor notes
|
|Student notes

Here are a few key points to summarize this section.

- Veracity is about trusting your data, and without veracity you can't expect to get good value from your data.
- For each data source, you need to discover its veracity, clean and transform what you can to improve it, and then prevent unwanted changes after you have cleaned it.

|Slide number 33
|Instructor notes
|
|Student notes
This section discusses considerations for data veracity and its impact on value.

Evaluating the veracity of data

Data engineering-focused questions:

- How is the source managed and maintained?
- Who generated the data, and who owns it?
- How often is the data updated?
- Do we need all fields and all records?
- Do audit trails exist?

Data science-focused questions:

- What methods were used to collect and process the data?
- Is this dataset the product of scientific research methods?
- What type of bias is likely part of this dataset?
- Does the data appear to be complete?
- How recent is the most recent data? Are there time gaps?

|Slide number 34
|Instructor notes
|
|Student notes
Let's look briefly at the activities that you might perform to address each of these categories. You will go deeper into these topics in other modules in the course.

To evaluate the state of the data, you must ask the correct questions and dig into the dataset. You will want to wear your data engineer's hat to think about the mechanics and technical requirements for the data and your data scientist's hat to really identify whether the dataset will be valuable even in its cleanest form.

Best practices for cleaning data

Define clean

Come to agreement on what clean looks like for a source, and consistently apply the definition.

Trace errors

Don't just fix errors as part of your cleaning process—return to the source or sources and find the cause.

Change thoughtfully

Know what acceptable changes look like.

Ensure that cleaning processes don't make assumptions about what a data value means.

Retain auditable data

Don't discard raw data after cleaning if the raw data has business value.

|Slide number 35
|Instructor notes
|
|Student notes

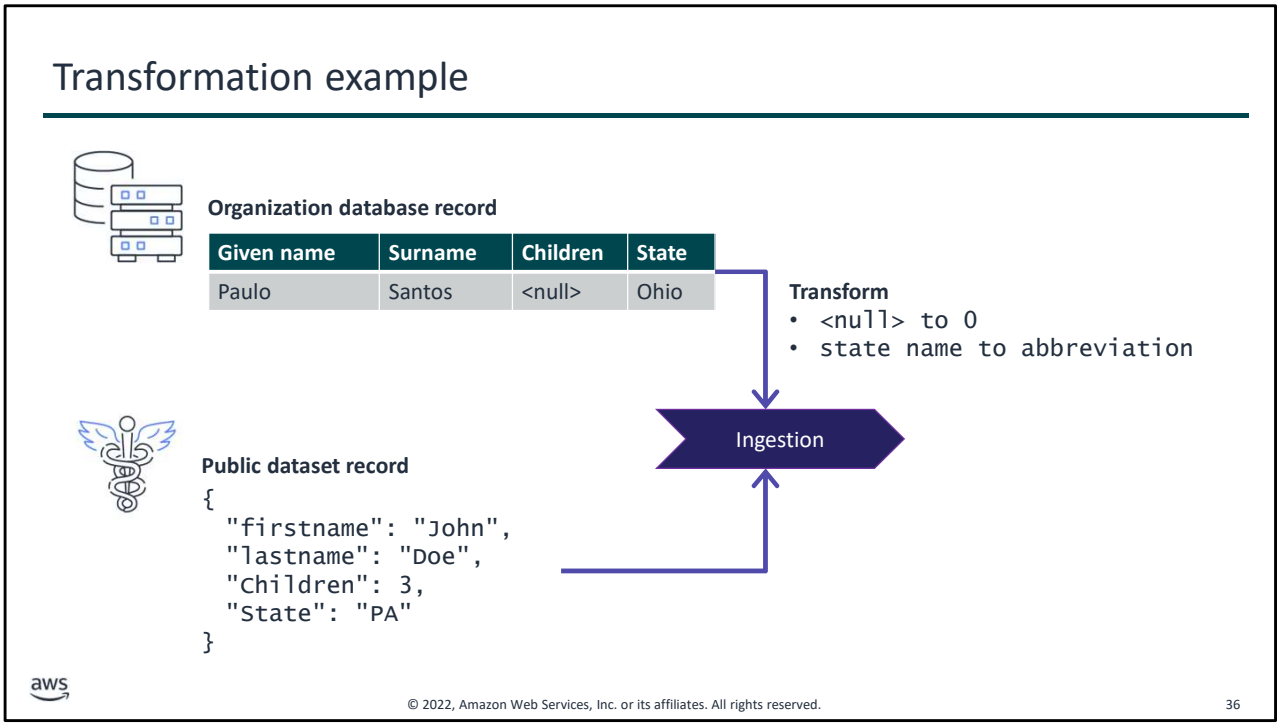
When it comes to cleaning your data, the first step is to come to a consensus on what clean looks like. Some businesses deem clean data to be data in its raw format with limited business rules applied. Others might consider data to be clean only after it has been normalized, been aggregated, and had value substitutions applied to regulate all entries. These are two very different understandings of clean. Be sure to know which one you are aiming for.

As you find errors in the data, trace them back to their likely source. This will help you with lineage and to predict workloads that will have integrity issues. It will also help you make a case for changes to an upstream system to improve the efficiency of ingesting the dataset into your pipeline on a recurring basis.

When you do need to change values in a field to use it, make sure that you understand the impact. For example, from a purely data-centric view, entering a zero in an empty column might seem like an easy data cleansing decision to make, but

<null> might not mean zero in the source that you are using. Similarly, data fields that sound like the same thing might actually be different.

In some systems, the original data is no longer valuable after it has been cleaned and transformed. However, with highly regulated data or highly volatile data, it's important to maintain both the original data and the transformed data in the destination system. For example, in an online gaming system, there might not be value in recording every direction shift that a player makes as they move on the map. The only important value is when the player enters or exits key areas of the map. However, in a banking app, all details of every transaction are vital for auditing, even though a customer might only care to see if their transaction was successful or not.



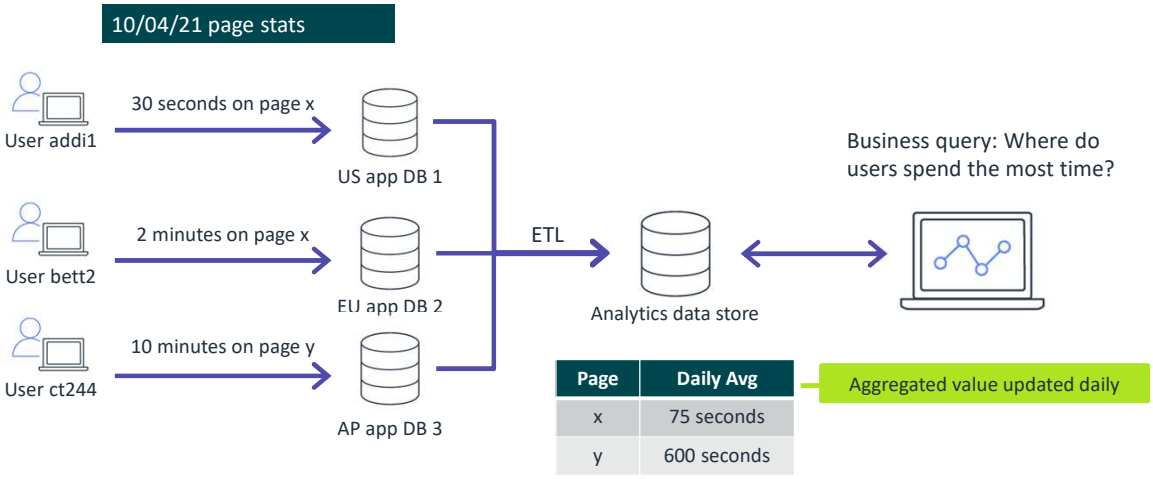
|Slide number 36
|Instructor notes
|
|Student notes

Even if each data source has been carefully maintained, sources will have differences that you need to address with transformations. Transformations are likely to occur across the ingest and processing phases, and the types of transformations might be quite diverse.

Transformations can be basic, such as the example pictured on this slide where you need to convert field values to a common format. In this example, the database record is transformed to replace the null value with a zero in the Children field and replace the word Ohio with OH in the State field.

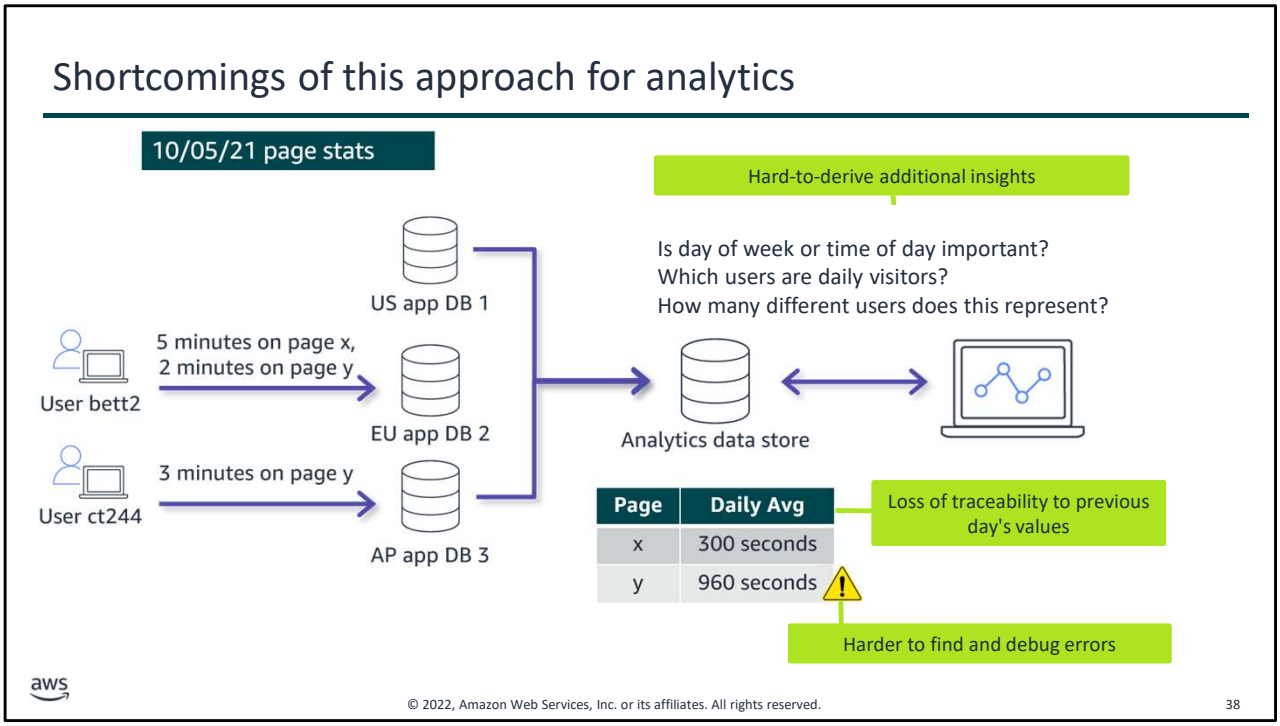
Transformations can also be more advanced, such as applying business rules to the data to calculate new values. Advanced transformations include filtering records, complex join operations, aggregating rows, splitting columns, and data validation.

Saving aggregated data compared to saving raw data records



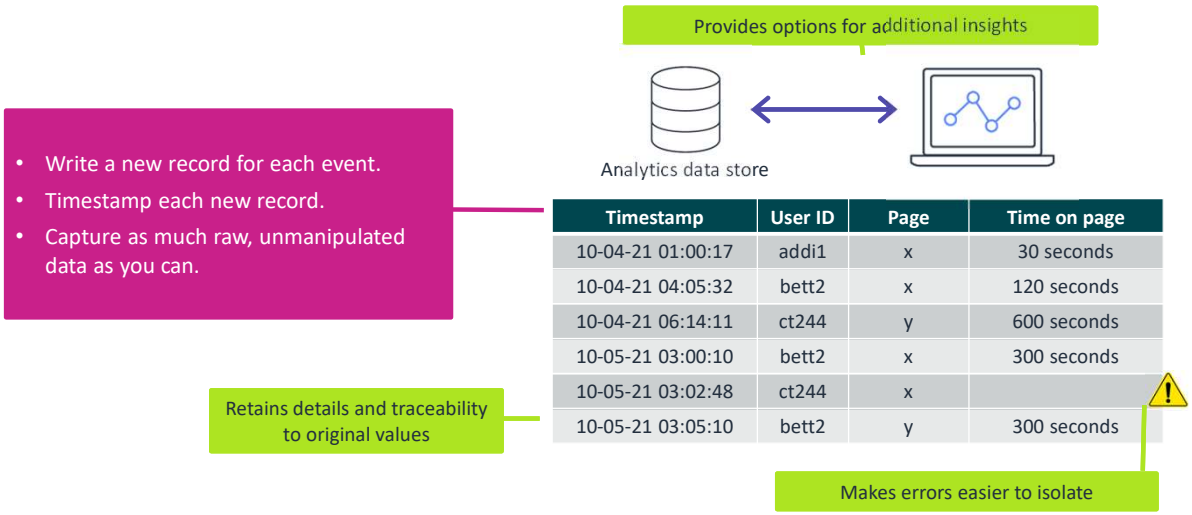
|Slide number 37
|Instructor notes
|
|Student notes

A common approach with applications that aggregate data is to make updates directly to the aggregated value—for example, a count or average. Reporting and business intelligence (BI) tools query the database and have access to the aggregated value. In this example, the business is interested in learning where users spend most of their time while on the website. To support this, their extract, transform, and load (ETL) process ingests page view stats daily and updates the daily average for each page in their analytics database.



|Slide number 38
|Instructor notes
|
|Student notes
As the slide indicates, there are shortcomings to the approach of saving only the aggregated values. First, this approach is not flexible if the business decides that they want to dig deeper or ask different questions. Second, the analytics database doesn't have any traceability back to the previous day's value after the new day's data has been incorporated into the average. Finally, this approach makes it more difficult to find and debug errors. In this example, the total of 960 seconds for page y is incorrect, but there isn't a way to easily figure out what is causing the error or to audit it.

Better approach for analytics: Immutable data



Timestamp	User ID	Page	Time on page
10-04-21 01:00:17	addi1	x	30 seconds
10-04-21 04:05:32	bett2	x	120 seconds
10-04-21 06:14:11	ct244	y	600 seconds
10-05-21 03:00:10	bett2	x	300 seconds
10-05-21 03:02:48	ct244	x	
10-05-21 03:05:10	bett2	y	300 seconds

|Slide number 39
|Instructor notes
|
|Student notes

This slide highlights a better approach for analytics to support the value and veracity of data. The ETL process writes a row in the analytics data store for each page view and includes a time stamp. This gives the business the opportunity to query the data in different ways and retains the original values so that queries could be done for different time periods. This approach also provides traceability to the original values and makes it easier to debug problems when errors occur. In this example, something went wrong when capturing the value for October 5 for one of the users. But, because you have the details for each day, user, and webpage, you can identify that data is missing. By using timestamps and capturing as much unmanipulated data as you can, you provide flexibility and traceability in your analytics solutions.

Maintaining data integrity and consistency

- Secure all layers of the pipeline.
- Grant least privilege access.
- Apply best practices to maintain data integrity.
- Keep audit trails.
- Implement data compliance and governance processes including data classification and data cataloging.
- Maintain a single source of truth.

|Slide number 40
|Instructor notes
|
|Student notes
Now that you have clean, trustworthy data in your pipeline, it's your job to keep it that way. This means preventing unwanted changes, knowing what has changed, and putting rules and processes in place that limit the opportunity for issues to be introduced.

You need to secure your pipeline layers as data moves through them and when data is stored in them. Use the best practice of granting least privilege access to any users who should be allowed access to the data. For example, many users will only require read-only access.

Different types of data stores will have different methods to maintain data integrity, and you need to implement the recommended best practices for each type. It is also important that you implement audit trails so that you can trace back any changes.

At an organizational level, the organization should implement an overarching data

compliance and governance process that applies across all of the data stores and access methods. The governance process should include classifying data according to its security level and cataloging data to make it findable. An important part of this approach is maintaining a single source of truth for an element of data. This means that one data store is the system of record, and any other data stores that use that data load it from the primary source.

Key takeaways: Activities to improve veracity and value

- You need to ask questions about the trustworthiness and lineage of the source data.
- To clean data, you need a common definition and must avoid making assumptions about what values should be.
- Data transformations can be simple, such as value substitutions, or complex, such as deriving new values.
- Save timestamped details instead of aggregated values.
- The organization should implement compliance and governance strategies to protect the integrity of data in your systems.

|Slide number 41
|Instructor notes
|
|Student notes

Here are a few key points to summarize this section.

- As a data engineer or data scientist, you need to question the data source before it ever enters the pipeline.
- After you determine to use a data source, you need to determine how to clean it and have a common definition of what clean means for a data source. Ensure that any transformation does what you intended.
- Save raw data in an immutable form so that you have details instead of only aggregated values. This supports future insights, and makes it easier to find errors.
- To protect your cleaned data, your organization should implement processes and governance strategies to manage the data in your systems.

Activity: Planning Your Pipeline

- In this activity, you will select a scenario and document your ideas about planning a pipeline for the scenario. Use the 5 Vs of data to guide you.
- Use the detailed instructions and the worksheet provided in your online course to complete this activity.

|Slide number 42
|Instructor notes
|The goal of this activity is to give students example business use cases and ask them to think about the data questions in terms of the Vs, the types of data, and what might be needed to transform the data.
|Students can choose an example and work through it. There are no correct answers—this is intended to help learners to recognize the factors that must be considered. There are suggestions in the instructor's answer key to help prompt discussions.
|The instructor files section of the online course provides detailed instructions and a worksheet you can use in the PipelinePlanningActivity.pdf file. This is suggested as a group breakout activity, but it could also be done individually with results shared within the online course. The use cases introduced here will be revisited in later modules of the course. The fraud use case is revisited in the Processing Data for ML module, The gaming analytics use case is revisited in the Analyzing and Visualizing Data module, and the e-bike use case is revisited in its own PowerPoint deck titled IoT Use Case Slides. This deck is intended as supplemental learning across the nodules focused on ingestion, storage, and analysis and visualization of IoT data in a simplified scenario. You might choose to use it as reinforcement at the close of each

related module, or as a review after all modules have been completed.
|
|Student notes
Throughout the course, the use cases that are introduced in this activity will be used to illustrate concepts and provide a means to compare different design choices.

|Slide number 43
|Instructor notes
|
|Student notes
This section summarizes what you have learned and brings the Elements of Data module to a close.

Module summary

This module prepared you to do the following:

- List the five Vs of data.
- Describe the impact of volume and velocity on your data pipeline.
- Compare and contrast structured, semistructured, and unstructured data types.
- Identify data sources that are commonly used to feed data pipelines.
- Pose questions about data to assess its veracity.
- Suggest methods to improve the veracity of data in your pipeline.

|Slide number 44
|Instructor notes
|
|Student notes
The module introduced a basic vocabulary to think about the data sources that will feed your pipeline. You learned about volume, velocity, variety, veracity, and value and how each of them impacts your pipeline design. You also learned the importance of asking questions about data sources before they enter your pipeline and protecting them after they are part of your system.

Module knowledge check

- The knowledge check is delivered online within your course.
- The knowledge check includes 10 questions that are based on material from the slides and slide notes.
- You can retake the knowledge check as many times as you would like.

|Slide number 45
|Instructor notes
|
|Student notes
Use your online course to access the knowledge check for this module.

Sample exam question

A data engineer has been asked to develop a pipeline to analyze partner sales trends by using sales transaction data. Each sales partner provides a .csv file each month. Although the files contain similar data, they are not in a standard format.

Which option is the most effective approach to collect this data for a data analytics pipeline?

Identify the key words and phrases before continuing.

The following are the key words and phrases:

- Using **sales transaction data**
- Monthly .**csv file** from **partners**
- Similar data, **not in a standard format**
- **Collect** the **data**

|Slide number 46
|Instructor notes: The key words section is animated to be revealed on click.
|
|Student notes
The question notes that the pipeline will use sales transaction data that is received from multiple sales partners. The data that needs to be collected into the pipeline is not in a standard format.

Sample exam question: Response choices

You have been asked to develop a pipeline to analyze partner sales trends by using **sales transaction data**. Each partner provides a **.csv file each month**. Although the files contain similar data, they are **not in a standard format**.

Which option is the most effective approach to collect this data for a data analytics pipeline?

Choice	Response
A	Extract the number of sales from each partner file. Then, write the partner ID and sales total to the analytics data store for analysis.
B	Extract the data from each partner's file, and transform it into a uniform file structure before writing it to the analytics data store for analysis.
C	Identify sources of statistical bias as part of extracting and transforming the data for analysis.
D	Import all of the files into the analytics data store with a partner ID. Grant access to each partner to edit the records and fix errors in only their data.

aws

|Slide number 47
|Instructor notes
|
|Student notes
Use the key words that you identified on the previous slide, and review each of the possible responses to determine which one best addresses the question.

Sample exam question: Answer

The correct answer is B.

Choice	Response
B	Extract the data from each partner's file, and transform it into a uniform file structure before writing it to the analytics data store for analysis.

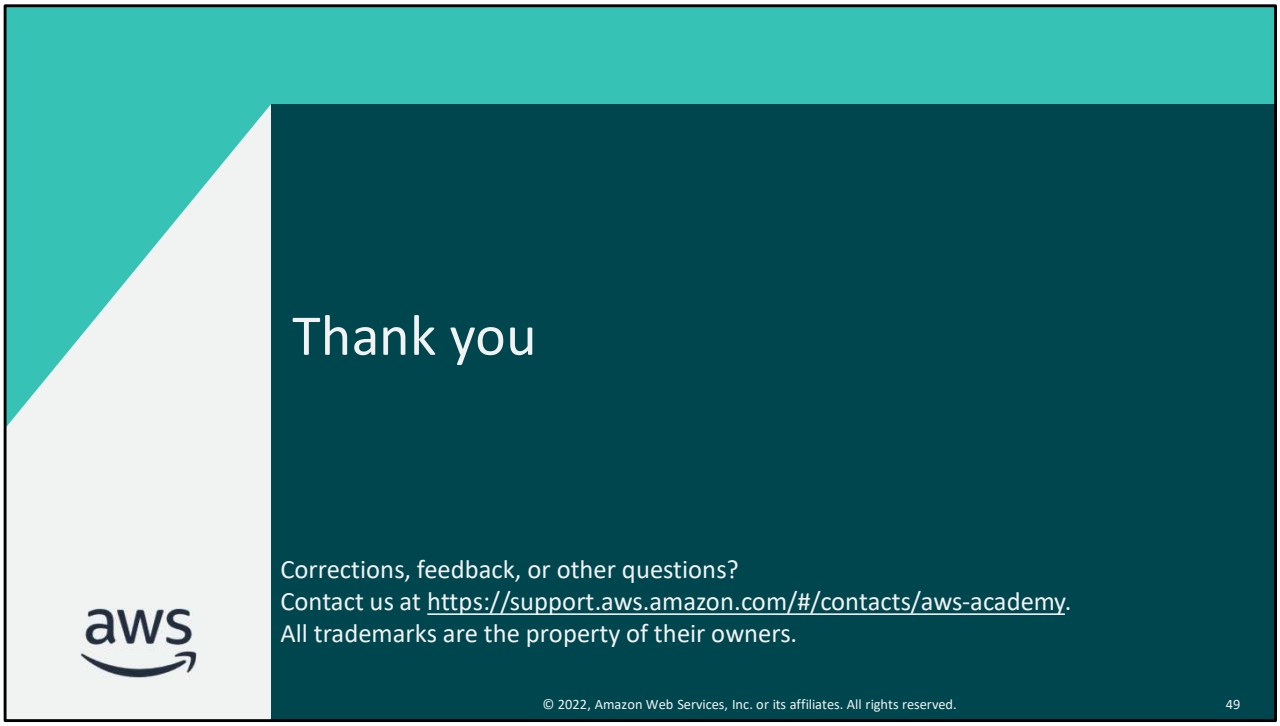
|Slide number 48
|Instructor notes
|
|Student notes
To use the data from partner files in the pipeline, you need to define a unified structure and transform each partner file into the common structure, so B is the correct answer.

Choice A (Extract the number of sales from each file, then write the partner ID and sales total to the analytics data store for analysis) does describe a way to combine the data and record summary sales data, but the best practice would be to write the detailed record to the analytics store for both future analysis and auditability.

Choice C (Identify sources of statistical bias as part of extracting and transforming the data for analysis) is not a relevant consideration for sales transaction data. It also isn't a process that you would perform as part of collecting data from its source and bringing it into your pipeline.

Choice D (Import all files into the analytics data store with a partner ID, then grant

access to each partner to edit the records and fix errors in only their data) isn't a good approach because it goes against best practices to protect the veracity of data in your pipeline. You might ask partners to fix errors in their source databases if appropriate, but you don’t want people modifying the data in your analytics store. Avoid giving write access to your analytics store to reduce the chance for human error.



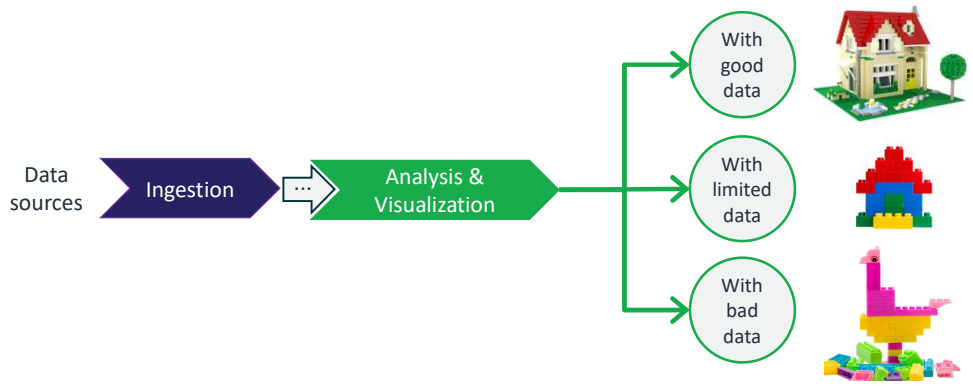
|Slide number 49
|Instructor notes
|
|Student notes
That concludes this module. The Content Resources page of your course includes links to additional resources that are related to this module.

Scaling your pipeline for volume and velocity of data

- Amount of data plus pace of data drives design choices.
- Volume and velocity impact all layers of the pipeline.
- Each pipeline layer must be evaluated for its own requirements.
- Choices should balance costs for throughput and storage against the required time to answer and accuracy of the answer.

Value rests on veracity

Making a data-driven decision with bad data is worse than making a decision without data.



|Slide number 51

|Instructor notes

|This is a light-hearted analogy, but it might be worth calling out the distinctions it aims to show. With good data, you get a well defined model with lots of details realized. If you have less data, but it is still good quality data, you can still derive the concept of a house from the data, just with much less clarity. If you attempt to process the model with bad data, your outcome is something completely different (in this case a chicken).

|

|Student notes

It might seem obvious, but it's important to remember that the impact of bad data can be worse than working without much data. When you work with limited data, decision makers are aware of the limitations and can manage risk accordingly. If consumers trust bad data to drive decision-making, the results could be dramatically different.

Shortcomings of this approach for analytics