Event-driven

Build and extend event-driven solutions on cloud-native architectures.

← Architecture Center (/cloud/garage/architectures)

Event-driven

Process continuous streaming events

Let’s talk

Overview

(/cloud/garage/architectures/eventDrivenArchitecture/overview)

Reference architecture

(/cloud/garage/architectures/eventDrivenArchitecture/reference-architecture)

Extended reference architecture

(architectures/eventDrivenExtendedArchitecture)

Get code examples

Event sources, cloud-native applications, and microservices

(/cloud/garage/architectures/eventDrivenArchitecture/1_0)

Related practices

Event storming

One of the essential elements of modern event-driven solutions is the ability to process continuous event streams to derive insights and intelligence. You can use specialized streaming analytics engines to run stateful analytical and complex event-processing workloads across event streams while they maintain low latency processing times.

By including these engines as part of the event-driven architecture, you enable these capabilities:

- Analysis and understanding of event streams
- Extracting event data from the stream so that data scientists can understand and derive machine learning models
- Running analytical processes and machine learning models against the event stream
- Matching of complex event patterns across streams and time windows to make decisions and take actions

Streaming analytics

Streaming analytics provides the capabilities to look into and understand the events that flow through unbounded event streams. Streaming applications process the event flow and

(/cloud/garage/architectures/eventDrivenArchitecture/event-storming-methodology)

Related learning

Event sources (/cloud/garage/architectures/eventDrivenArchitecture/event-driven-event-sources)

Event backbone (/cloud/garage/architectures/eventDrivenArchitecture/event-driven-event-backbone)

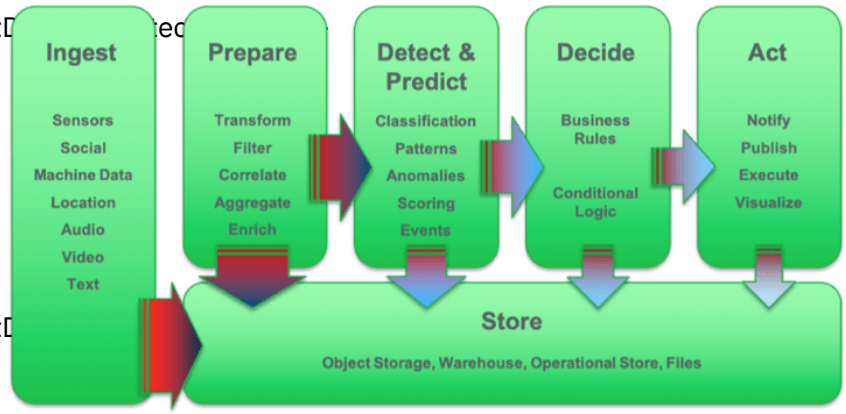
Act after an event with IBM Cloud Functions (/cloud/garage/architectures/eventD…ed-driven-take-action-with-cloud-functions)

Process event streams (/cloud/garage/architectures/eventD…driven-event-streams)

Event-managed state

and/or data and apply/call functions to be applied to information in the stream. These applications are written as multistep flows across these capabilities:

- Ingest many sources of events
- Prepare data by transforming, filtering, correlating, and aggregating on some metrics and by using other data sources for data enrichment
- Detect and predict event patterns by using scoring and classification
- Decide by applying business rules and business logic
- Act by directly running an action, or, in event-driven systems, publishing an event notification or command



Basic capabilities for streaming analytics

Let’s talk

(/cloud/garage/architectures/eventDrivenArchitecture/event-driven-event-managed-state)

Event-driven cloud-native applications (microservices)

(/cloud/garage/architectures/eventDrivenArchitecture/event-driven-cloud-native-apps)

To support the analytical processing of the unbounded event streams, these capabilities are essential to the event stream processing component:

- Continuous event ingestion and analytical processing
- Processing across event streams
- Low latency processing, where data doesn't need to be stored
- Processing of high-volume and high-velocity streams of data
- Continuous query and analysis of feeds
- Correlation across events and streams
- Windowing and stateful processing
- Query and analysis of stored data
- Development and running of data pipelines
- Development and running of analytics pipelines
- Scoring of machine learning models in line in the event-stream processing

Let’s talk

Support for analytics and decision-making

Beyond the basic streaming capabilities, consider supporting other frequently seen event stream types and processing functions in your event stream processing component. By creating functions for these stream types and processes in the streaming application code, you can simplify the problem and reduce the development time. This list contains examples of other event stream types and processes:

- Geospatial
  - Location-based analytics
  - Geofencing and map matching
  - Spatiotemporal hangout detection
- Time series analysis
  - Timestamped data analysis
  - Anomaly detection and forecasting
- Text analytics
  - Natural Language Processing (NLP) and Natural Language Understanding (NLU)

    - Sentiment analysis and entity extraction

- Video and audio
    - Speech-to-text conversion
    - Image recognition
- Rules: Decisions that are described as business logic
- Complex Event Processing (CEP): Temporal pattern detection
- Entity analytics
    - Relationships between entities
    - Probabilistic matching

Application programming languages and standards

Few standards exist for event-stream applications and languages. Typically, streaming engines provide language-specific programming models that are tied to a specific platform. The commonly used languages are as follows:

- Python supports working with data and is popular with data scientists and data engineers.

- Java™ is the pervasive application development language.

- Scala adds functional programming and immutable objects to Java.

Other platform-specific languages emerged as real-time processing demanded more stringent performance requirements. For example, Google started the Apache Beam project (https://beam.apache.org/) to provide a unified programming model for streaming analytics applications. Beam is a higher-level unified programming model that provides a standard way of writing streaming analytics applications in many supported languages, including Java, Python, Go, and SQL. Streaming analytics engines typically support this unified programming model through a Beam runner that takes the code and converts it to platform-native executable code for the specific engine.

For more information, see the details of supporting engines and capabilities (https://beam.apache.org/documentation/runners/capability-matrix/). Leading engines include Google Cloud Dataflow, Apache Flink, Apache Spark, Apache Apex, and IBM® Streams.

Runtime characteristics

In operational terms, streaming analytics engines must receive and analyze arriving data continuously:

- The Feed Never Ends:
  - The collection is unbounded.
  - The model isn't based on a request response set.
- The Fire Hose Doesn't Stop:
  - Keep drinking and keep up.
  - The processing rate is greater than or equal to the feed rate.
  - The analytics engine must be resilient and self-healing.

These specialized demands and concerns aren't found in many other information processing environments. These concerns led to highly optimized runtimes and engines for the stateful, parallel processing of analytical workloads across event streams.

Please log in () to send us your feedback!

Let's talk

Let IBM experts help you get started with your enterprise transformation.

Contact us

Learn

Topic (https://www.ibm.com/cloud/)

Data Centers (https://www.ibm.com/cloud/data-centers)

Products (https://www.ibm.com/cloud/products/)

Solutions (https://www.ibm.com/cloud/solutions/)

Build

Get started (https://www.ibm.com/cloud/garage/get-started)

Docs (https://console.bluemix.net/docs/?cm_mmc=IBMBluemixGarageMethod-_-MethodSite-_-10-19-15::12-31-18-_-docs)

Architectures (/cloud/garage/architectures)

Engage

Support (https://www.ibm.com/cloud/support)

Services (https://www.ibm.com/cloud/services)

Garage Home ()

Partners (https://www.ibm.com/cloud/partners)

Your Account

Sign up (https://console.bluemix.net/registration/?cm_mmc=IBMBluemixGarageMethod-_-MethodSite-_-10-19-15::12-31-18-_-bm_reg)

Login (https://console.ng.bluemix.net/login/?cm_mmc=IBMBluemixGarageMethod-_-MethodSite-_-10-19-15::12-31-18-_-bm_login)

Let’s talk

Customers (https://www.ibm.com/case-studies)

Security (https://www.ibm.com/cloud/security)

Practices (/cloud/garage/practices/practices)

Toolchains (/cloud/garage/toolchains)

Courses (/cloud/garage/category/courses)

(/cloud/garage/dte)

Get help (https://support.ng.bluemix.net/gethelp/?cm_mmc=IBMBluemixGarageMethod-_-MethodSite-_-10-19-15::12-31-18-_-bm_help)

Login issues (https://support.ng.bluemix.net/sup app=account&index=2&cm_mmc=IBMBluemix(_-MethodSite-_-10-19-15::12-31-18-_-bm_tic

Analytics (/cloud/garage/analytics)

Social

Blog (https://www.ibm.com/blogs/bluemix/)

Twitter (https://twitter.com/IBMCloud)

YouTube (https://www.youtube.com/user/IBMCloud)

Facebook (https://www.facebook.com/IBMCloud/)

Garage

Visit a Garage (/cloud/garage/what-is-the-cloud-garage)

Cloud adoption and transformation services (/cloud/garage/services)

What's New (/cloud/garage/news)

Garage Method for Cloud

Culture (/cloud/garage/practices/culture)

Discover (/cloud/garage/practices/discover)

Think (/cloud/garage/practices/think)

Develop (/cloud/garage/practices/develop)

Reason (/cloud/garage/practices/reason)

Operate (/cloud/garage/practices/operate)

Learn (/cloud/garage/practices/learn)

Let’s talk

Let’s talk