

Real-Time Games Analytics and Leaderboard with RisingWave, Kafka, and Superset (Preset)



Fahad Shah

Developer Advocate

Session Overview

Objective: Set up a real-time data pipeline to analyze and visualize player performance.

Scenario: Fortnite gaming data analysis

Data Set: Game statistics in matches of 400 players.

Key Tools:



For data ingestion and event streaming.



For real-time data processing and analytics.



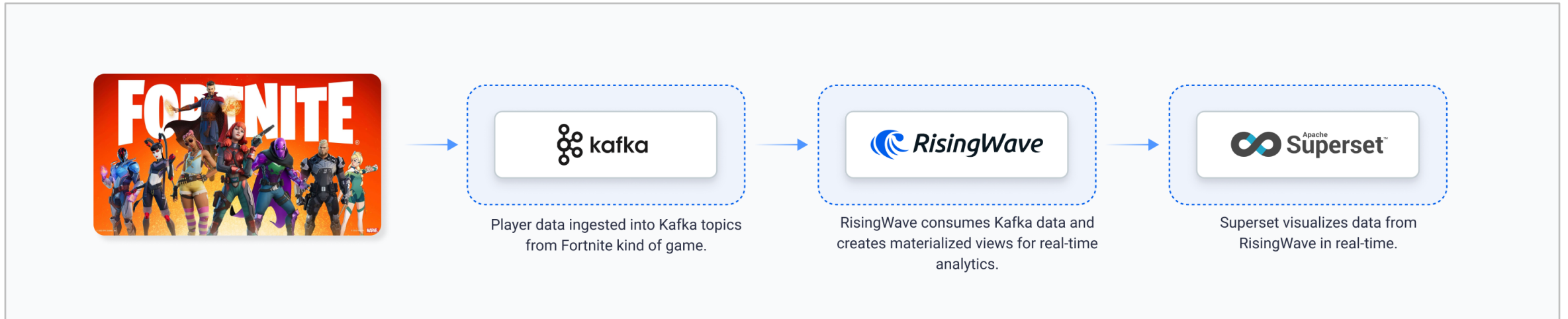
For data visualization and dashboard creation.



Pipeline Architecture

Architecture Diagram:

Visualize the end-to-end architecture of streaming pipeline



Data Ingestion with Kafka

Objective: Stream player-related data into a Kafka topic.

Data Details: Key fields for player metrics (e.g., *player_name*, *mental_state*, *match_id*, *total_kills*, *damage_dealt*, *distance_moved*, *kills*, *assists*, *player_id*, *timestamps*).

Kafka Setup:

Define a Kafka topic, e.g., *player_stats*.

Configure Kafka producer to simulate player data streaming.

Data is streamed continuously to mimic real-time player data.

Overview of RisingWave

RisingWave is a SQL data platform that is PostgreSQL compatible.

Designed for building:

- Event-driven applications
- Real-time ETL pipelines
- Continuous analytics services
- Feature stores for AI applications

Excels in extracting fresh and consistent insights from:

- Real-time event streams
- Database change data capture (CDC)
- Time series data

Achieves insights within sub-seconds

Unifies streaming and batch processing, allowing users to:

- Ingest, join, and analyze live and historical data.
- Operate at cloud scale with ease.



Setting Up RisingWave for Real-Time Analytics

Objective:

Use RisingWave to consume real-time Kafka data and create materialized views for insightful game data analysis.

Steps:

1. Connect RisingWave to Kafka:

Define a Kafka source in RisingWave to ingest streaming data.

2. Create Materialized Views:

Set up materialized views to aggregate key metrics, such as:

- Total kills, accuracy, ranks, and damage dealt.
- Top 3 players with the highest kills.
- Top 3 players with the highest average survival time.
- Top 3 players with the highest accuracy in each 30-minute window.

Benefits:

Use PostgreSQL-style SQL within RisingWave to efficiently extract and analyze insights from player game data.



Configuring Superset in Preset

Objective: Visualize RisingWave data in real-time dashboards.

Steps:

Connect Superset to RisingWave: Set up a RisingWave database connection.

Create Dashboards and Charts: Use materialized views for charts (e.g., leaderboard, match stats).

Example charts:

- Top players by kills.
- Total kills, accuracy, ranks, and damage dealt.
- Top 3 players with the highest kills.
- Top 3 players with the highest average survival time.
- Top 3 players with the highest accuracy in each 30-minute window.
- Real-time player activity over time.

Result: Interactive dashboards for tracking and analysis.



Building Visual Dashboards

Objective: Create actionable visualizations for player performance.

Example Dashboards:

Leaderboard: Top-performing players based on kills, assists, etc.

Game Trend Analysis: Player metrics over time.

Alerts Setup: Define thresholds (e.g., player streaks, sudden dips in performance).

Automate Reports: Schedule daily/weekly reports on player trends.

Use Case: Alerts notify game managers of significant player activities, like new high scores.

Benefit: Dashboards provide instant insights, helping to make game adjustments in real-time.

Key Benefits of This Real-Time Pipeline

Advantages:

Instant Analysis: Real-time player performance tracking.

Engagement Insights: Analyze trends to improve gameplay and engagement.

Automated Monitoring: Alerts and reports keep teams informed without manual checking.

Real-World Applications: Used in live online games, e-sports events, and user engagement analysis.

Summary and Next Steps

Recap:

Set up a Kafka topic for data streaming.

Used RisingWave' materialized views for real-time data processing and analysis.

Built visual dashboards and alerts in Superset.

Next Steps:

Experiment with new metrics and visualizations.

Expand the pipeline to include more game data types.

Optimize for scalability and add automated testing.

Thanks & Q&A

Join the RisingWave Community!



risingwave.com/slack



risingwave.com/github