



# Azure PlayFab 데이터 플랫폼으로 플레이어 경험 향상

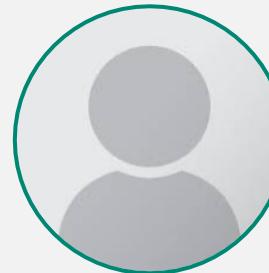
Nir Schleyen - He/Him, Satya Bobba - He/Him

# 소개



**Nir Schleyen**

수석 소프트웨어 엔지니어, He/Him



**Satya Bobba**

수석 소프트웨어 엔지니어링 T10, He/Him

# 안건

- PlayFab PlayStream
- PlayFab 수집 파이프라인
- 데이터 연결 - 새로운 서비스 제품
  - 지원 시나리오: 실시간, 일괄 처리, 임시 쿼리
  - 사용자 환경 워크스루
- 고객 사례 - Turn 10
- 활용 방안



# PlayStream

# PlayStream이란?

The screenshot shows the PlayStream Monitor interface. On the left is a world map with a grey oval highlighting the Americas and Europe. To the right is a list of events:

- processLogin revision 44 executed in 0.4043742s
- Player action executed
- Sent push notification with push notification template named test2
- Player triggered action executed Cloud Script SUCCESS: processLogin() (revision 46)
- ASF3 Entity logged in
- Sent push notification
- Player logged in custom

At the top, there are tabs for Overview and PlayStream Monitor, and buttons for Settings, PLAYSTREAM, Show all events, and Pause.

이벤트 파이프라인

The screenshot shows the Segment Editor. At the top, it says "Edit Segment" and "Segments > 1st Segment".  
Segment Information:

- Segment name: High XP Players (> 3000)
- Player count: 19

Player:

- Statistic value: Total\_XP\_Gained (is greater than 3000)
- Add Filter
- Add group

ACTIONS:

- Entered segment (1) Left segment
- Type: Devote Cloud Script
- Publish results as PlayStream Event
- Cloud Script Function: sendMessageToPlayer
- Arguments (JSON): { "title": "Congratulations!", "message": "You have unlocked new items in it!"}
- Remove action

실시간 세분화

The screenshot shows the "Edit Rule" dialog. It has tabs for Cloud Script, Rewarded Ads, Rules, and Scheduled Tasks. The Rules tab is selected.  
Name: Award item On Login  
Event type: com.playtika.player\_logged\_in  
Conditions:

- Add group

ACTIONS:

- Type: Devote Cloud Script
- Publish results as PlayStream Event
- Cloud Script Function: processLogin
- Arguments (JSON): {}

Save action Cancel

실시간 규칙

The screenshot shows the Data Explorer. At the top, it says "Data Explorer (basic)" and "Data in storage: 100 days | Data in cache: 14 days".  
Query:

- Start date: 01/04/2022
- End date: 01/10/2022
- Group by: Player Name
- Aggregate type: Count

Count:

- Group by: Player Name
- Aggregate type: Count
- Legend: blue = user\_region\_unlocked (1), green = user\_state\_changed (2), purple = character\_state\_changed (1), red = player\_state\_changed (2)
- Chart: Bar chart showing count for each player ID from 1 to 18.

Event name: user\_region\_unlocked  
Timestamp (UTC): 21/05/2022 00:00:00  
Player ID: >  
Event date: >

이벤트 기록

# PlayStream의 작동 방식

- 기술

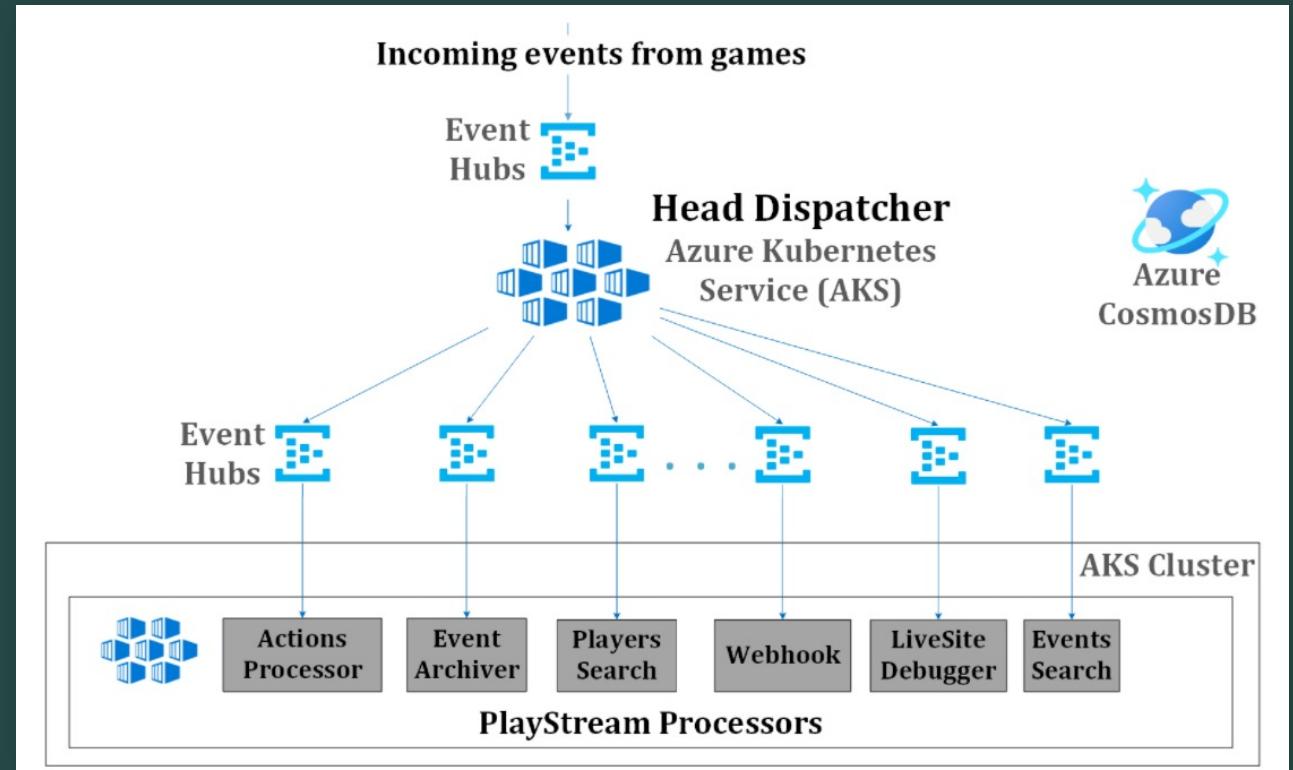
- Event Hubs
- AKS(Azure Kubernetes Service)

- 지원/보증

- 한 번 이상 제공
- 표준 및 맞춤형 이벤트
- 샘플링

- 기타 서버 백업

- 원격 분석
- 실험
- 플레이어 검색 및 세분화





# 데이터 연결

# 데이터 연결

## 데이터 연결이란?

- 다음과 같은 고객 리소스의 통합 지점:
  - 스토리지
  - ADX(Azure Data Explorer)
  - Event Hubs
  - 기타 등등

## 작동 방식

- 지속적으로 데이터 배포
  - 선택한 대상
  - 권한을 부여받은 대상
- 실시간
  - PlayStream + 원격 분석
  - Parque(BLOB용)
  - 기본 제공 재시도(최대 24시간)

# 데이터 연결의 발전



# 데이터 연결: 사용 사례

장기간 보존을  
위해 데이터를  
보관하고자 함

- 새로운 Azure Storage 계정(+컨테이너) **프로비전**
- PlayFab 권한 부여
- 새 컨테이너에 대한 쓰기 구성

Synapse에서  
자체적으로  
보고하기를 원함

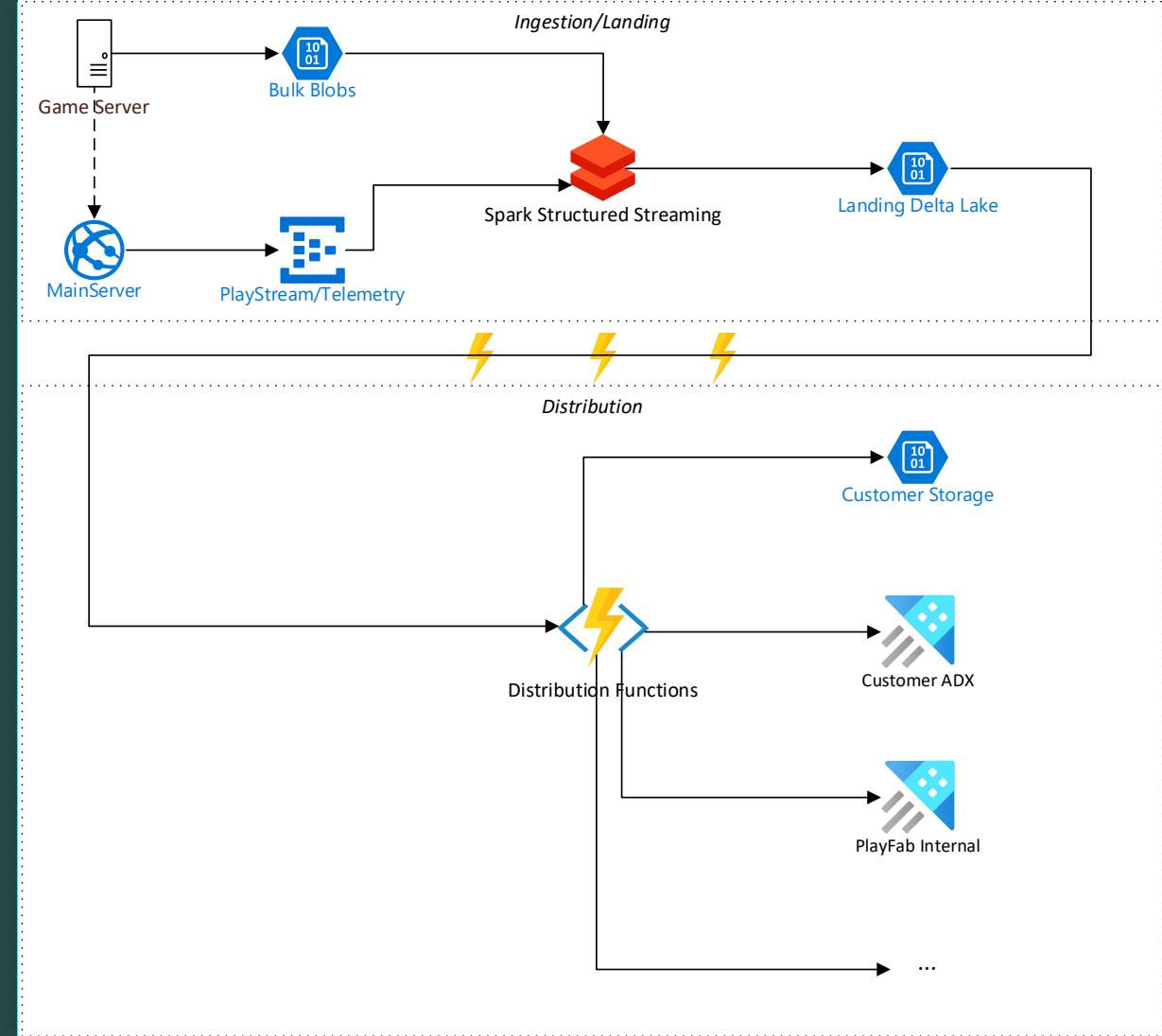
- 새로운 Azure Storage 계정(+컨테이너) **프로비전**
- PlayFab 권한 부여
- 새 컨테이너에 대한 쓰기 구성

임시 쿼리를  
실행하려고 함

- 새로운 Azure Data Explorer 데이터베이스 **프로비전**
- PlayFab 권한 부여
- 새 DB에 대한 PlayFab 쓰기 구성

# 데이터 수집 파이프라인

- PlayStream의 확장  
(수집 + 배포)
- 원격 분석 지원
- 비용 절감 - 데이터 재분할
- 높은 처리량
- 5분 미만의 대기 시간
- 데이터 연결
  - Azure Storage(Parque 형식)
  - 추가 기능 제공 예정



# 데이터 연결

## Blob Storage 연결 구성

The screenshot shows the Microsoft Game Data Explorer interface. At the top, there's a navigation bar with a back arrow, a search bar containing 'My Game', and three tabs: 'Data Explorer (basic)', 'Data Explorer (advanced)', and 'Webhooks'. Below the navigation is a sidebar with sections: 'Development' (dropdown), 'Title Overview' (0/100K), 'BUILD' (Players, Multiplayer, Groups, Automation, Add-ons), 'ENGAGE' (Economy, Leaderboards, Content), and 'ANALYZE' (Dashboards, Data, Experiments). The main content area is titled 'Data Connections' under 'Data Connections > New Data Connection'. It has a 'DATA DETAILS' section with fields for 'Name\*' (empty), 'AzureResourceId\*' (empty), 'DestinationType' (set to 'AzureBlobStorage'), 'AccountName\*' (empty), 'ContainerName\*' (empty), and 'TenantId\*' (empty). At the bottom are 'Save' and 'Cancel' buttons.

# 이벤트 샘플링

- 비용과 정확도 간 균형 유지
- 이벤트별 정의 제어
- 맞춤형 + 표준 이벤트
- 분석 정확도 향상을 위한 스마트/엔터티 기반 샘플링
- 수집 기반 샘플링(배포 이전)
  - 모든 데이터 연결에 적용됨
  - PlayStream에는 적용되지 않음

The screenshot shows the 'Manage Events (Preview)' tab in the Azure Data Explorer. At the top, there are tabs for Data Explorer (basic), Data Explorer (advanced), Webhooks, Event Archive, Event Export, Management (Preview), and Manage Events (Preview). Below the tabs, a message states: "Manage events. Each event stored in Azure Data Explorer will result in charges. Storing only a sample of events can reduce costs, but may also reduce reporting accuracy." The main area is divided into two sections: "Sampled events" and "Add sample rate".

**Sampled events:**

Event name	Event namespace	Category	Sample rate
character_created	com.playfab	standard	100 % No sampling
Client_BattleAborted	title.ASF3	standard	50 %
client_focus_change	playfab.sessions	standard	50 %

**Add sample rate:**

Event namespace: title.ASF3  
Event name: Client\_BossKill  
Sample rate: 50 %

A dropdown menu for 'Event name' lists several options: Client\_BossKill, Client\_LevelComplete, Client\_LevelUp, Client\_PlayerDied, Client\_RegisteredAccount, Client\_SaleClicked, and Client\_StoreVisit. A 'Save' button is located at the bottom right of the dialog.



# 고객 사례/데모

Turn 10 팀



# Turn 10: Azure PlayFab 데이터 플랫폼 기반

- 플레이어 경험 향상
  - 대기 시간 탐지
  - 크래시
- 플레이어 동작 이해
  - 플레이어 진행
  - 클릭 수
- 게임 개발
  - 게임 품질 개선
  - A/B 테스트



# Turn 10: Azure PlayFab 데이터 플랫폼 기반

- 현황
  - ADF + 맞춤형 ETL
  - ADX(Azure Data Explorer) 기반
- 제한 사항
  - ADX에 대한 대규모 데이터 세트 분석
  - 종종 시간 제한 초과 발생
  - 제한 사항을 해결하기 위해 추가 엔지니어링 작업 필요



# Turn 10: Azure PlayFab 데이터 플랫폼 기반

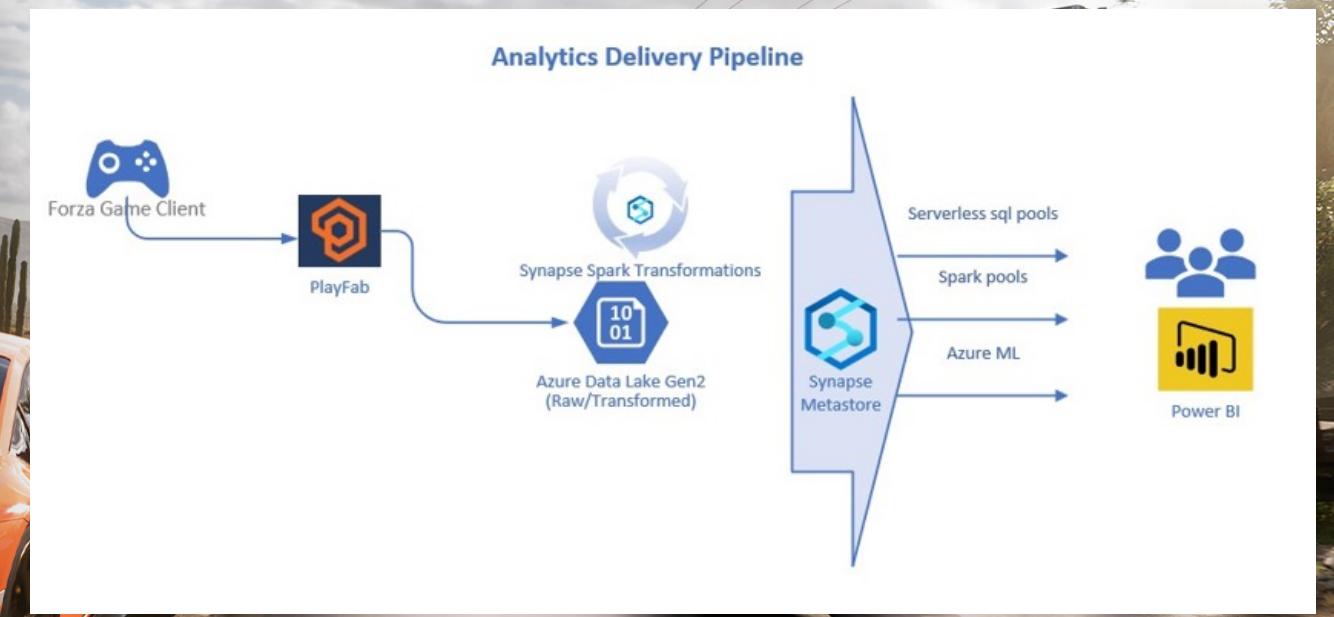
- 초기 계획(BYOS 이전)
  - ADX 전문가
  - Spark를 사용한 Synapse Analytics
- 제한 사항
  - ADX에 추가 비용 소요
  - 추가 대기 시간
  - 또다른 실패 지점/흡





# Turn 10: Azure PlayFab 데이터 연결 기반

- ADLS Gen2
- Spark를 사용한 Synapse Analytics
- 대규모 데이터 세트에서 셀프 서비스 수행
- 근 실시간
- 비용 절감



# 리소스

기술 리소스 참조:  
데이터 연결 및 이벤트 샘플링

<https://docs.microsoft.com/en-us/gaming/playfab/features/data/manage-events-with-sampling/>

추가 질문이 있는 경우 포럼 방문:  
[aka.ms/playfabforums](http://aka.ms/playfabforums)

API 리소스 참조:

<https://docs.microsoft.com/en-us/rest/api/playfab/experimentation/experimentation?view=playfab-rest>

무료 계정으로 지금 시작:  
[aka.ms/playfabaccount](http://aka.ms/playfabaccount)

