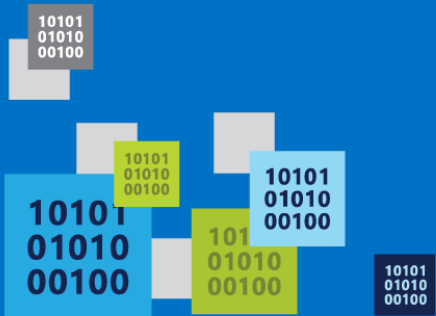


# Azure 데이터 저장소 (Data Storage)

Dae Woo Kim (daewoo.kim@)  
Technology Evangelist  
Microsoft Korea

<https://aka.ms/azure-camp-dec>



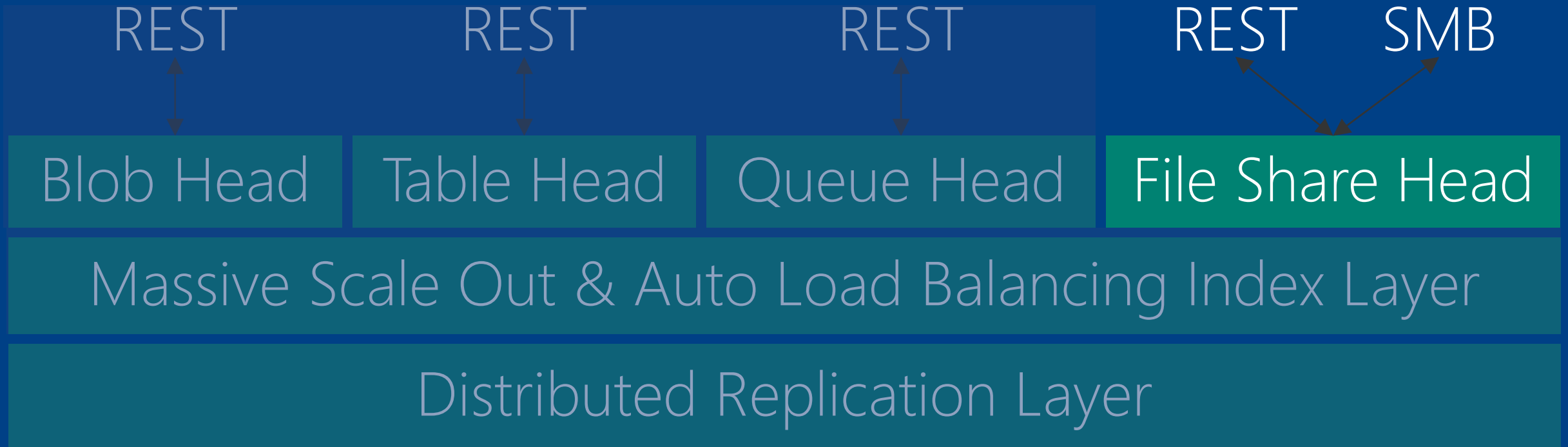
# 목차

- Blobs
- Files
- Queues

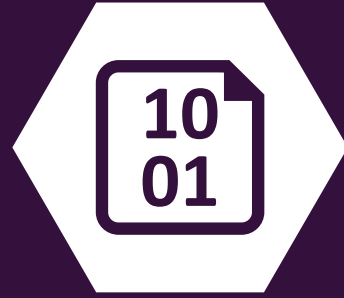
- Tables
- StorSimple

# Azure 저장소 아키텍처

Microsoft Azure



# Blobs



# Microsoft Azure Storage Blob

# 두가지 종류의 Blob

Block Blob

Page Blob

# Block Blob

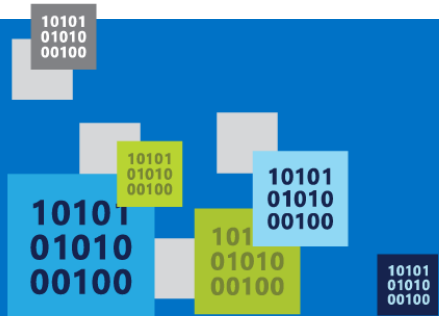
스트리밍 작업(streaming workloads)에 적합

개별 blob은 연속적인 block들로 구성됨

개별 block은 block id로 식별됨

Blob당 200GB 크기 제한

Etags 를 통해 Optimistic Concurrency 구현



# Page Blob

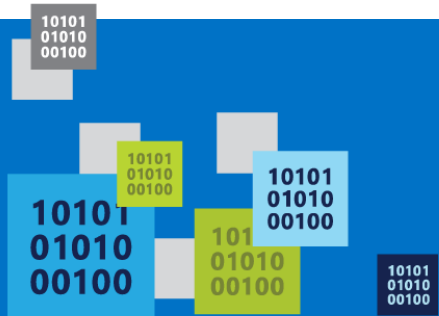
랜덤 (random read/write workloads) 작업에 적합

개발 blob은 page의 배열로 구성

개별 page는 blob의 시작부터 offset으로 식별됨

Blob당 1TB 크기 제한

Lease를 통해 Optimistic 또는 Pessimistic (locking) concurrency 구현됨

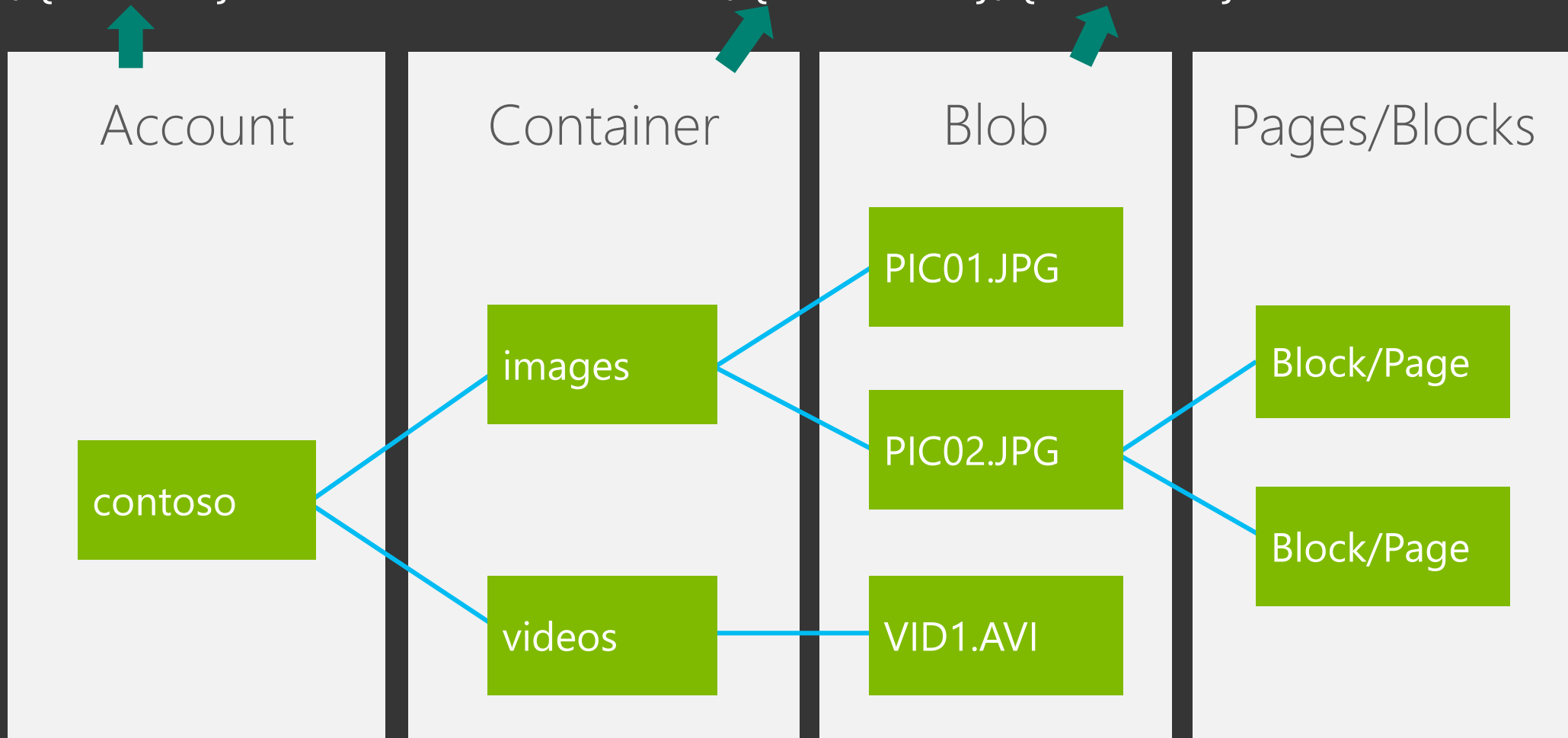




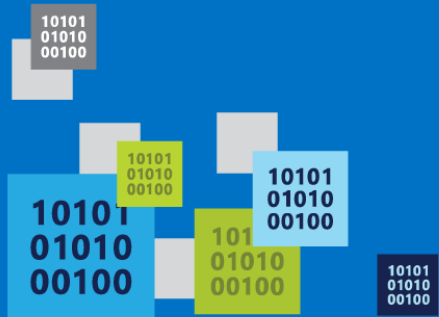
# Blob 저장소 구조

Microsoft Azure

`http://{account}.blob.core.windows.net/{container}/{blobname}`

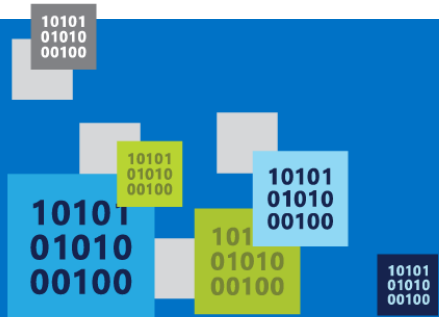


# Demo: blob과 통합



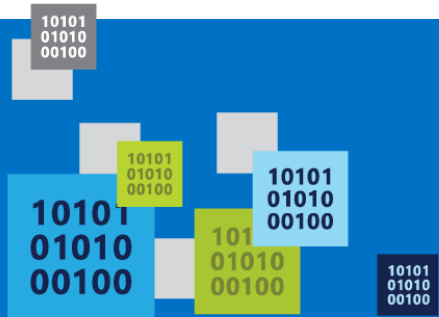
# Containers

- 계정(Account) 당 여러개의 Container
- 특수 목적의 \$root container



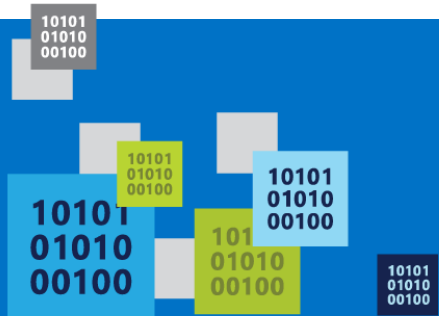
# Containers

- Container는 여러 세트의 blob을 보관
- 접근 정책(access policies)은 container 레벨로 제공
- Container와 함께 metadata가 보관됨
- Blob들의 목록이 container에 존재



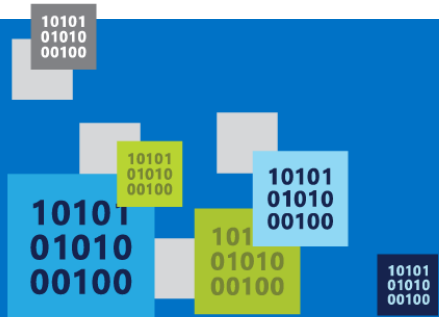
# 대역폭(Throughput)

- 파티션당 제공
- Blob당 60MB/s 제공

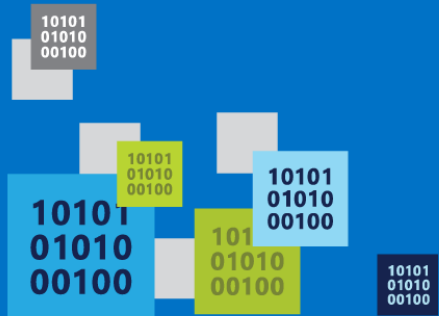


PutBlob  
GetBlob  
DeleteBlob  
CopyBlob  
SnapshotBlob  
LeaseBlob

Blob Details – Main Web Service Operations



# Demo: blob 저장소

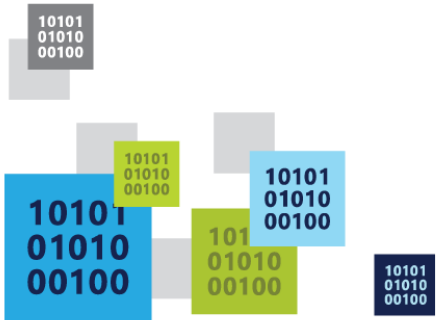


# Blob은 항상 이름으로 접근

Can include '/' or other delimiter in name

e.g. /<container>/myblobs/smurf.png

  
blob name





# Blob 리스트 샘플

```
http://adventureworks.blob.core.windows.net/  
  Products/Bikes/SuperDuperCycle.jpg  
  Products/Bikes/FastBike.jpg  
  Products/Canoes/Hybrid.jpg  
  Products/Canoes/Flatwater.jpg  
  Products/Canoes/Whitewater.jpg  
  Products/Tents/PalaceTent.jpg  
  Products/Tents/ShedTent.jpg
```

GET <http://.../products?comp=list&prefix=Tents>

```
<Blobs>  
  <Blob> <Name> Tents/PalaceTent.jpg </Name> [...] </Blob>  
  <Blob> <Name> Tents/ShedTent.jpg </Name> [...] </Blob>  
</Blobs>
```

# Blob 리스트 샘플 전체 응답 response

<Blobs>

<Blob>

<Name>Tents/PalaceTent.jpg</Name>

<Url>https://readinesscloudcamp.blob.core.windows.net/products/Tents/PalaceTent.jpg</Url>

<LastModified>Wed, 17 Dec 2014 09:00:26 GMT</LastModified>

<Etag>0x8D1E7EF08F31520</Etag>

<Size>150027</Size>

<ContentType>image/jpeg</ContentType>

<ContentEncoding />

<ContentLanguage />

</Blob>

<Blob>

<Name>Tents/ShedTent.jpg</Name>

<Url>https://readinesscloudcamp.blob.core.windows.net/products/Tents/ShedTent.jpg</Url>

<LastModified>Wed, 17 Dec 2014 09:00:26 GMT</LastModified>

<Etag>0x8D1E7EF08EA6257</Etag>

<Size>150027</Size>

<ContentType>image/jpeg</ContentType>

<ContentEncoding />

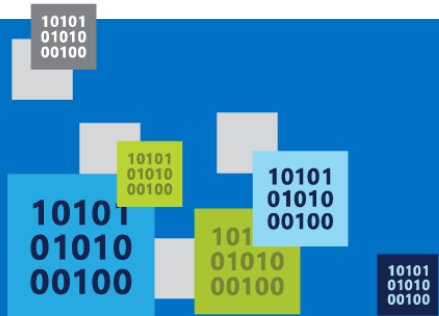
<ContentLanguage />

</Blob>

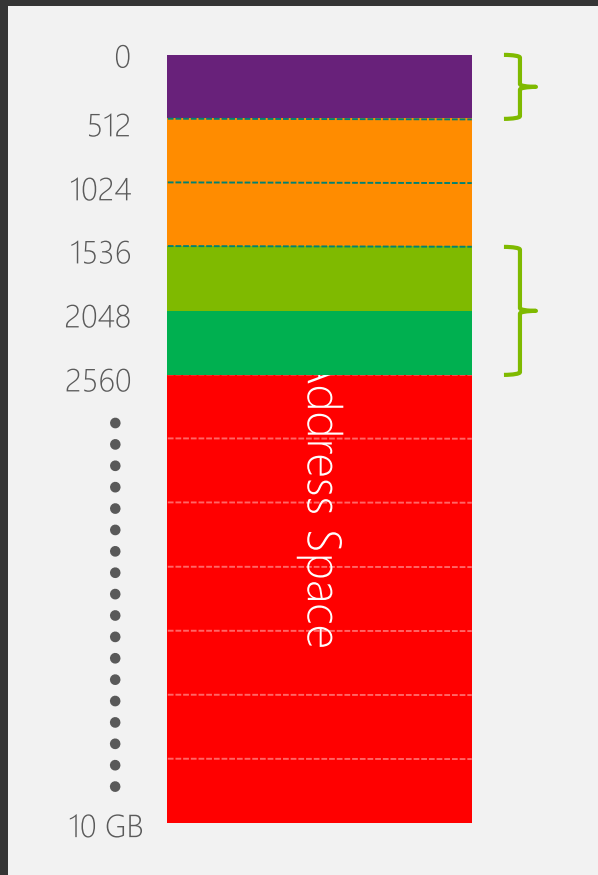
</Blobs>

# Blob block 업로드 혜택

효율적인 연속 업로드와 재시도  
병렬 및 순서와 무관한 block 업로드



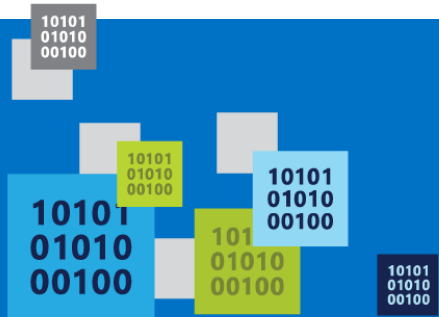
# Page Blob – 랜덤 Read/Write



Sparse 저장소:  
오직, page에 데이터가 저장될때  
비용이 부과됨  
(Only charged for pages with data  
stored in them)

# Shared Access Signatures

Blob과 containers 접근 제어를 위한 방안



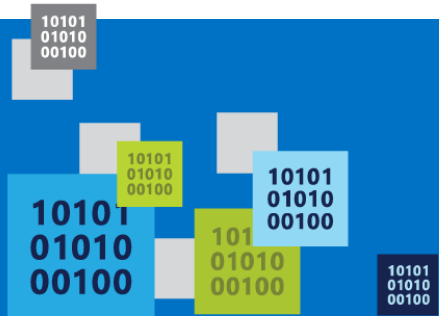
# Shared Access Signatures – 두가지 접근 방안

Ad-hoc:  
Stored Access Policy

정책기반:  
Shared Access Signature

# Shared Access Signatures – Revocation

짧은 기간에 사용하고 재발행  
삭제할 수 있는 container 레벨 정책에 적용



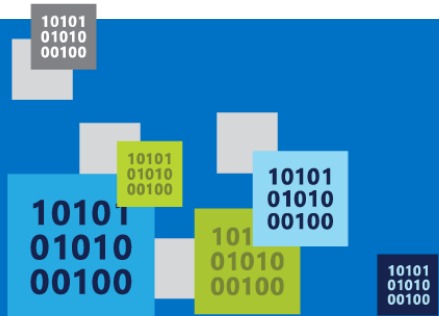
# Shared Access Signatures – Ad Hoc Signatures

## 짧은 기간의 Shared Access Signature 생성

Blob 또는 Container에 사인

AccessPolicy Start, Expiry와 Permissions

필드에 HMAC-SHA256 사인



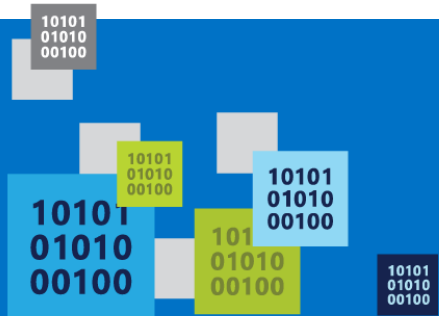


# Shared Access Signatures – Ad Hoc Signatures

적용 사례

URL에 1회 적용

E.g. 모바일 클라이언트에게 container 에 업로드를 위한 URL을 제공



# Shared Access Signatures

## Ad Hoc Signatures

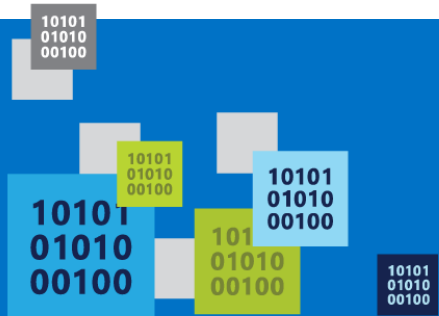
Microsoft Azure

```
http://...blob.../pics/image.jpg?  
sr=c&st=2009-02-09T08:20Z&se=2009-02-10T08:30Z&sp=w  
&sig= dD80ihBh5jfNpym05Hg1IdiJIEvHcJpCMiCMnN%2fRnbI%3d
```

# Store Access Policy – 정책 기반 Signatures

## Container 레벨 정책 생성

StartTime, ExpiryTime, Permissions 을 지정



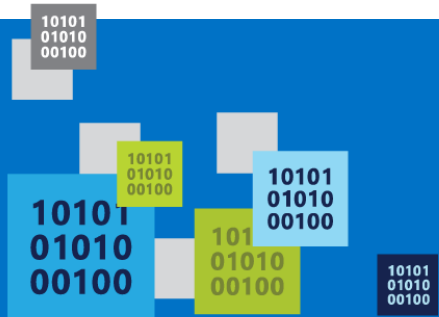
# Store Access Policy – 정책 기반 Signatures

## Shared Access Signature URL 생성

사인된 blob이나 container

사인된 identifier Optional pointer to container 정책

Signature HMAC-SHA256 of above fields

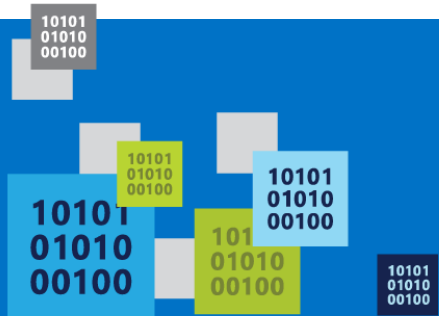


# Store Access Policy – 정책 기반 Signatures

## Use case

회수 가능한 권한을 특정 사용자나 그룹에 부여

revoke: container 정책을 수정하거나 삭제

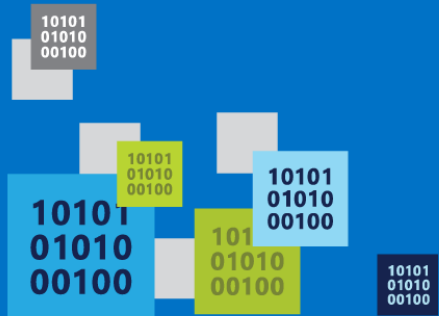


# Store Access 정책 정책 기반 Signatures

Microsoft Azure

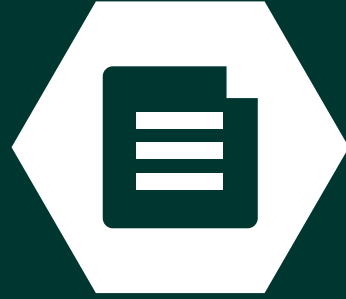
```
http://...blob.../pics/image.jpg?  
sr=c&si=MyUploadPolicyForUserID12345  
&sig=dD80ihBh5jfNpym05Hg1IdiJIEvHcJpCMiCMnN%2fRnbI%3d
```

# Demo: Shared Access Signatures



# Files





# Microsoft Azure Storage Files

Setup an IaaS VM to host a File Share backed by an IaaS Disk

Write code to find the IaaS File Share from the rest of the VMs in



Write s

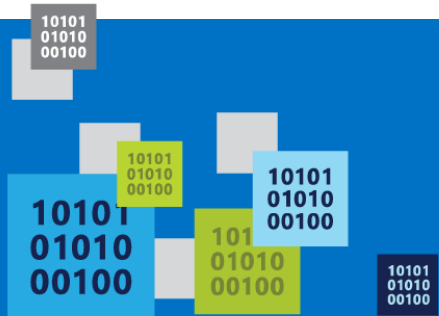
IaaS VM  
(Sharing IaaS Disk)

Backup IaaS VMs  
(Mount/Share after failover)

Handle most upgrades, node failures

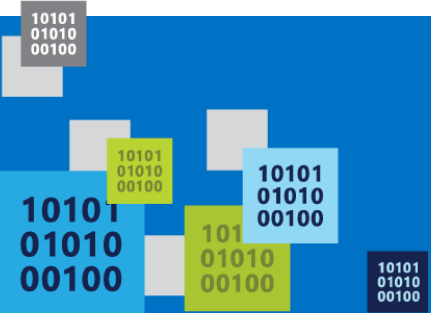
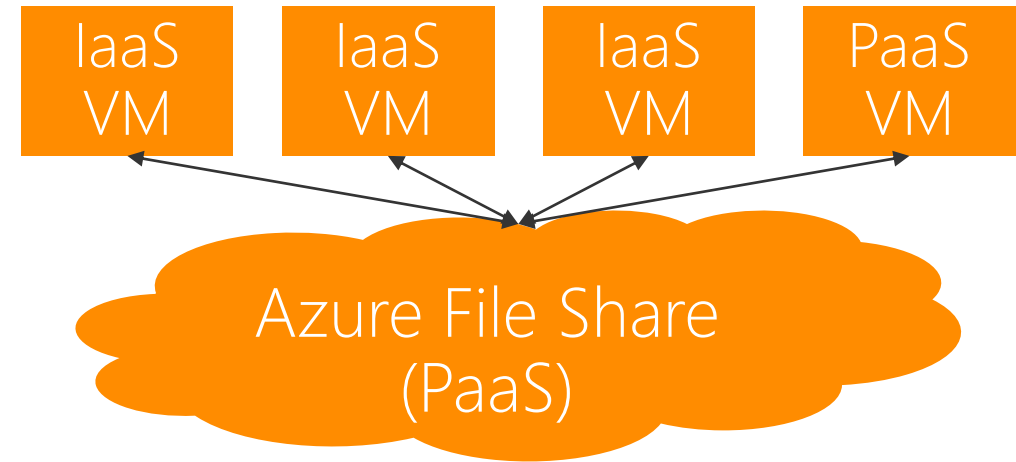
You can only access the File Share from other VMs

Sharing Files –  
예전 방식



# Azure Files

- Azure의 공유된 네트워크 파일 저장소
- 가용성, 안정성, 확장성과 관리되는 자동화 서비스
- 두개의 인터페이스 지원: SMB, REST

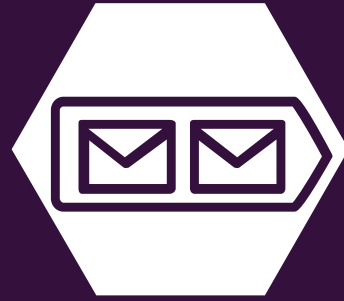


# Azure Files – 적용예

Microsoft Azure

- VM과 어플리케이션간 공유하는 데이터
- 서비스들의 설정 공유
- 개발/테스트/디버깅

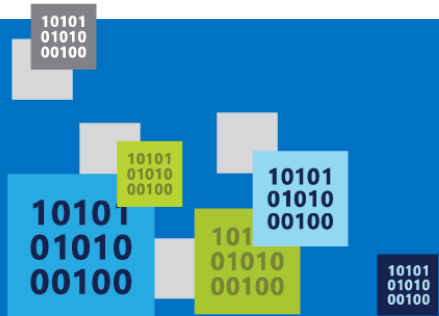
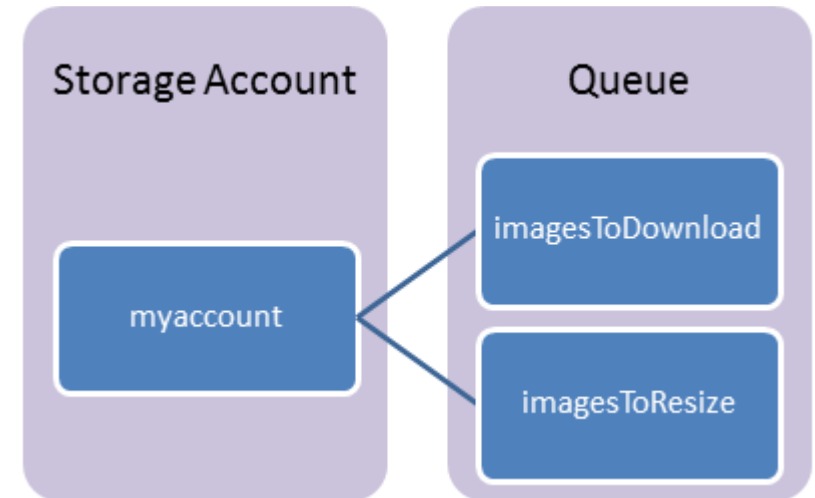
# Queues



# Microsoft Azure Storage Queue

# Queue 구성요소

- 저장소 계정 : 모든 Azure 저장소 접근은 저장소 계정을 통해 처리
- Queue: queue는 여러개의 message를 소유
- Message: message는 64KB까지 저장 가능한 포맷



# Queue URL 형식

Queue는 아래의 URL 형식으로 접근 가능

`http://{storage-account}.queue.core.windows.net/{queue}`

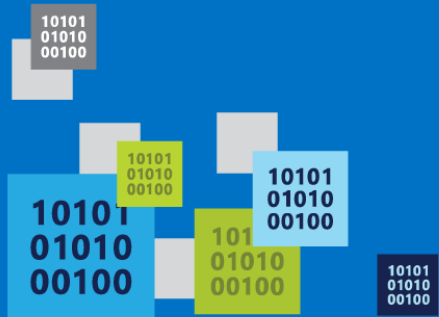


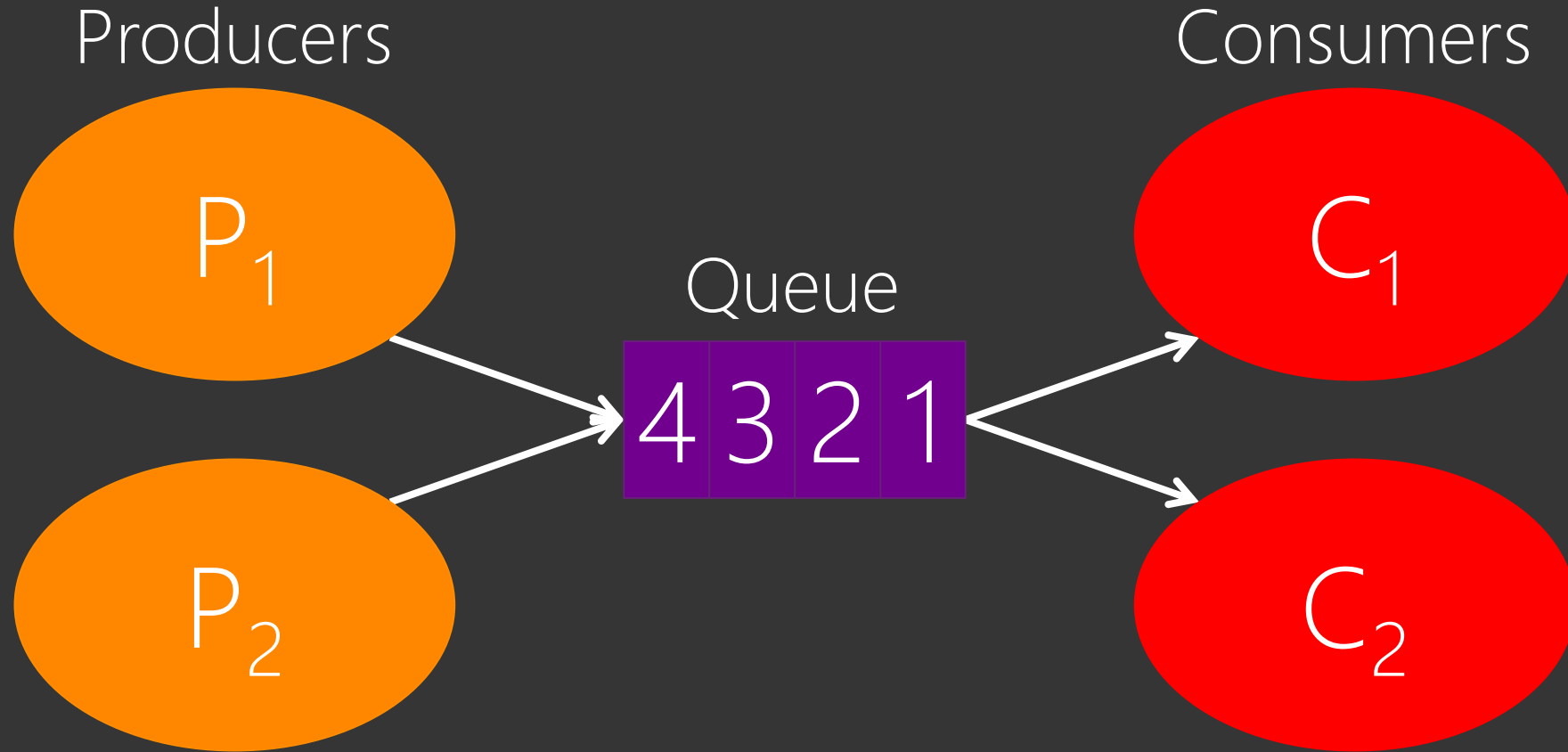
# Queue URL 형식

Example:

<http://myaccount.queue.core.windows.net/imagesToDownload>

# Demo: 비동기 처리를 위한 웹 어플리케이션





Message들은 순차처리되나 FIFO를 보장하지는  
않음

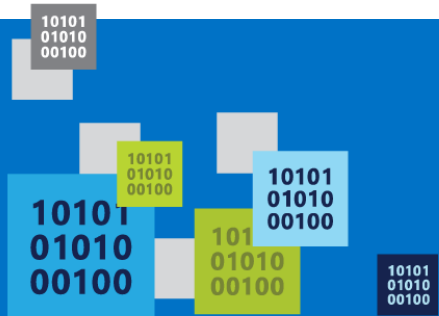
Message는 적어도 한번 처리되어야 함

Message는 여러번 처리될 수 있음

.DequeueCount 가 매번 증가

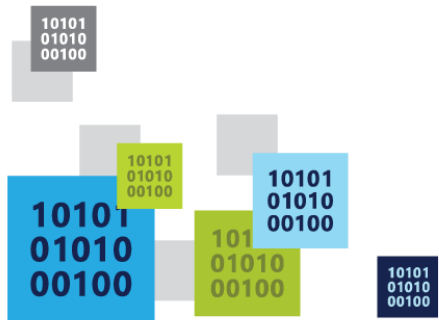
-> DequeueCount 처리가 중요함

## Queue Considerations

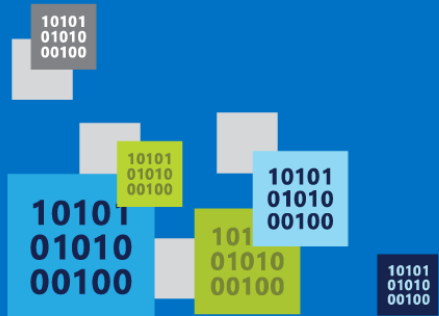


# Queue 고려사항

Message들은 7일간 저장됨



# Demo: Queues in Code



# Tables

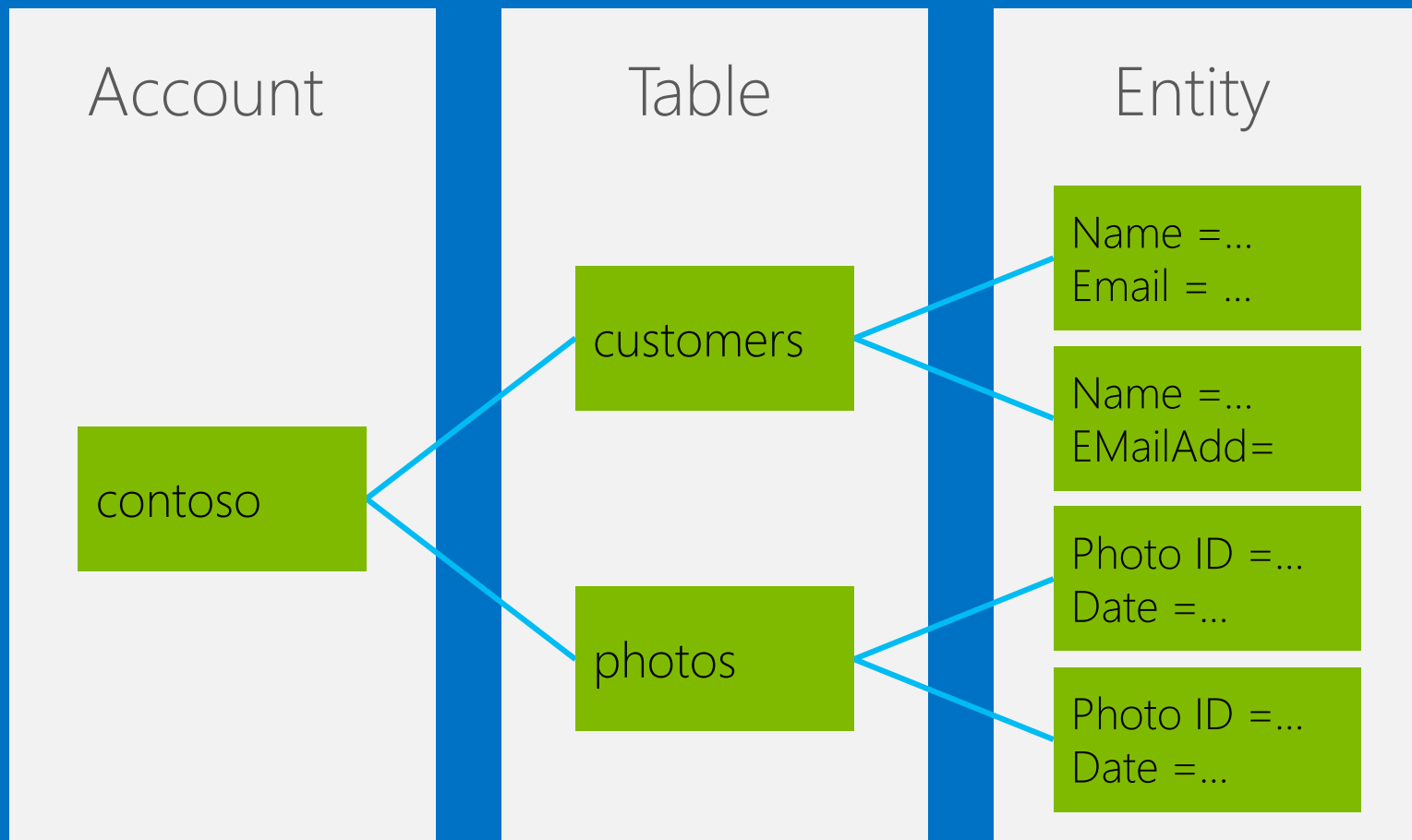


# Microsoft Azure Storage Table



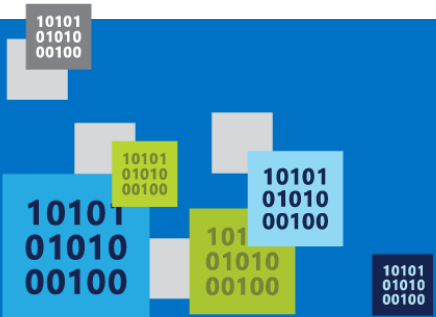
# Table 저장소 구조

Microsoft Azure



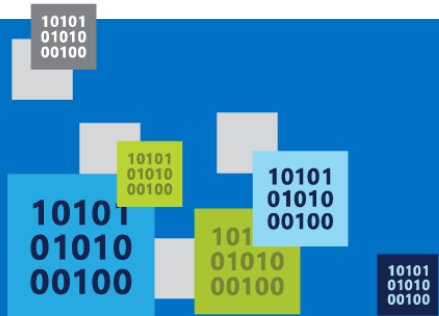
# Table 저장소

Not an RDBMS Table!  
'Entities'가 주요한 컨셉



# Table 저장소

Entity는 255개의 property를 보유 가능  
개별 entity 당 1MB 저장 가능



# Table 저장소

Microsoft Azure

## Entity Properties

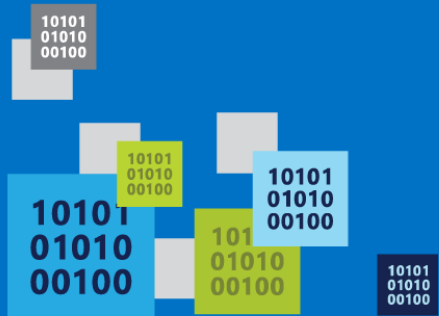
PartitionKey & RowKey 는 필수 Properties

복합키가 entity를 고유하게 식별

유일하게 색인된 property

정렬 순서를 정의

# Demo: table



# Table 저장소

Microsoft Azure

## PartitionKey의 목적

**Entity Locality**

**Entity Group Transactions**

**Table Scalability**

# Table 저장소

Microsoft Azure

## PartitionKey의 목적

### Entity Locality

같은 partition의 entity는 함께 저장됨

효율적인 쿼리와 지역 캐시 가능

모든 쿼리에 가능한 partition key를 넣어 쿼리하는게

유리함

# Table 저장소

Microsoft Azure

## PartitionKey의 목적

### Entity Group Transactions

같은 partition에 위치할 경우 여러 CRUD를 하나의 transaction으로 처리 가능



# Table 저장소

Microsoft Azure

## PartitionKey의 목적

### Table Scalability

처리량 – 500 tps/partition 및 7,000 tps/account

Azure가 table 저장소 사용 패턴을 모니터링

# Table 저장소

Microsoft Azure

## PartitionKey의 목적

### Table Scalability

파티션들에 대한 자동 부하 조절

개별 파티션은 다른 저장소 노드로 제공될 수 있음

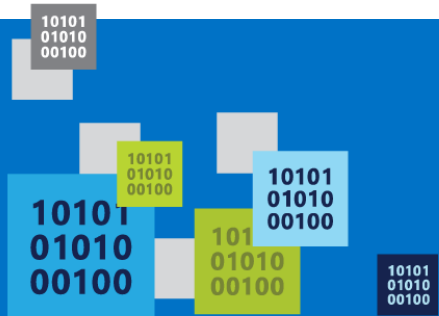
트래픽 요구를 충족할 수 있도록 테이블을 확장 가능

# Table Storage Details Entity Properties

Timestamp property

Optimistic Concurrency

HTTP Etag로 노출됨



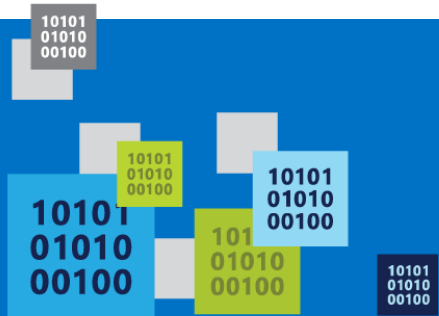
# Table Storage Details Entity Properties

다른 property에는 고정된 스키마 없음

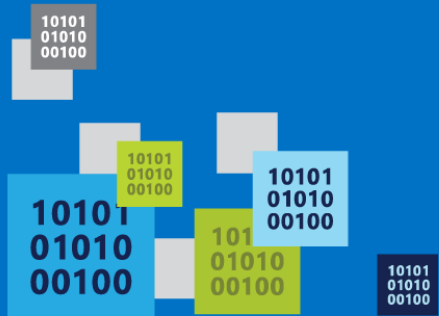
개별 property는 <name, typed value>로 저장됨

Property는 표준 .NET 형식:

string, binary, bool, DateTime, GUID, int, int64, double



# Demo: Enter "data" with varying shape into a table

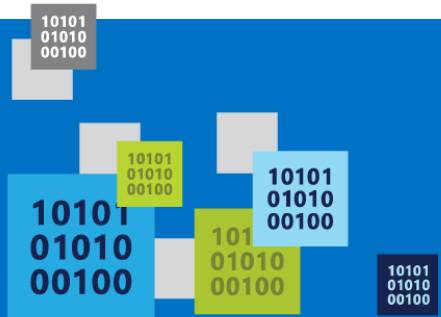


# Table 저장소

CRUD 지원

Upsert와 Entity Group Transactions 처리 포함

Table들은 metadata 보유



# StorSimple



# Microsoft Azure StorSimple



# StorSimple + Microsoft Azure = Hybrid Cloud Storage



# StorSimple

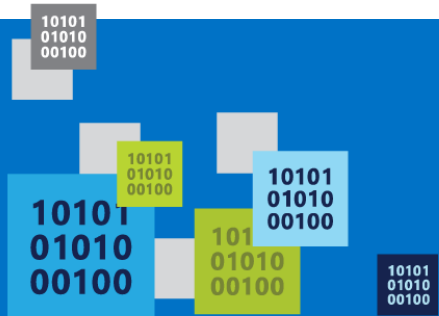
## Designed to:

저장소 비용 절감

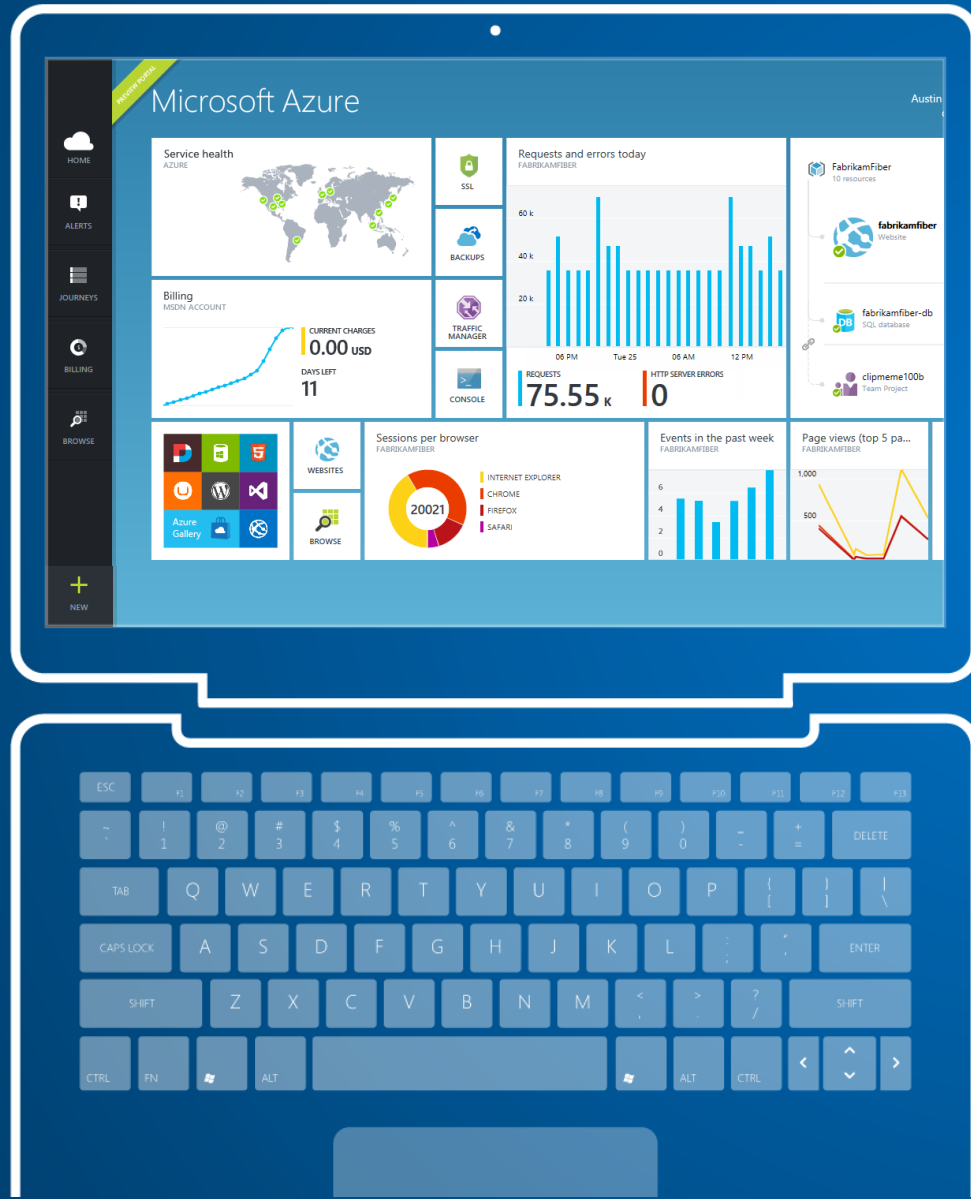
저장소 관리 단순화

재난 복구 기능 향상 및 효율성 증대

데이터 이동성 제공







Get started  
Visit [azure.microsoft.com](https://azure.microsoft.com)



© 2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.