# Globally Distributing Data



Leonard Lobel
CTO, SLEEK TECHNOLOGIES
lennilobel.wordpress.com



## Replication - Why?

#### Performance

Ensures high availability within a region

Across regions, brings data closer to the consumer

### **Business continuity**

In the event of major failure or natural disaster



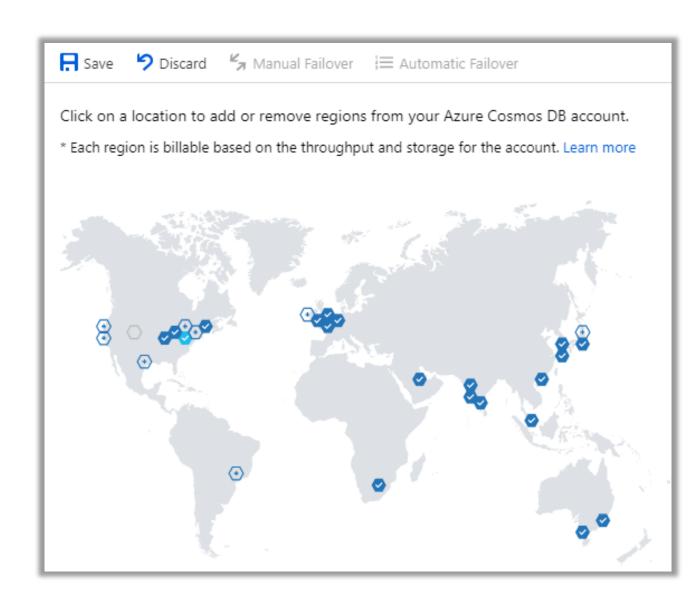


# Associate any number of regions with your Cosmos DB account

- Limited to geo-fencing policies

### Dynamically add/remove regions

- Associate (and disassociate) regions with the click of a mouse



# Associate any number of regions with your Cosmos DB account

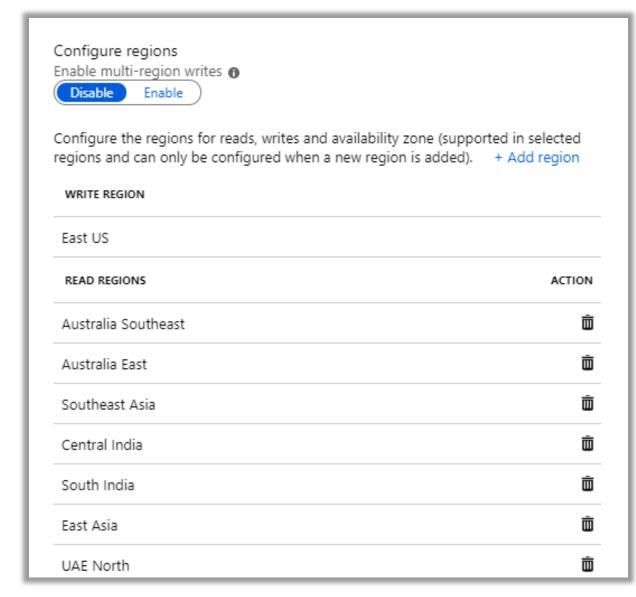
- Limited to geo-fencing policies

### Dynamically add/remove regions

- Associate (and disassociate) regions with the click of a mouse

#### Multi-master

- Enable writes across all regions, with automatic failover



# Associate any number of regions with your Cosmos DB account

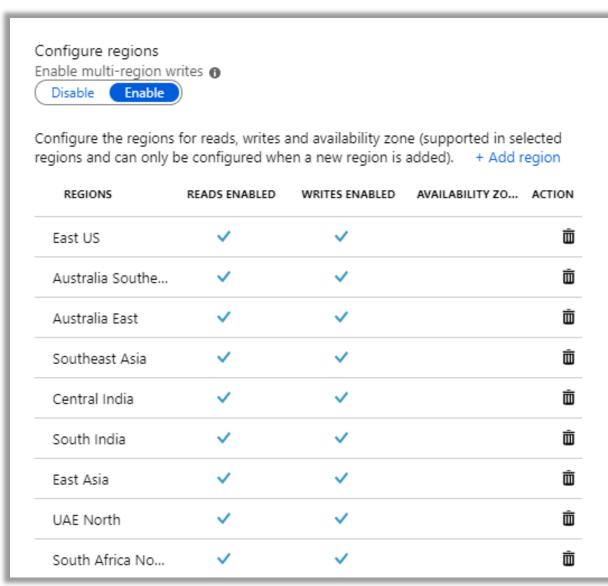
- Limited to geo-fencing policies

### Dynamically add/remove regions

- Associate (and disassociate) regions with the click of a mouse

#### Multi-master

- Enable writes across all regions, with automatic failover



# Demo



**Global distribution** 

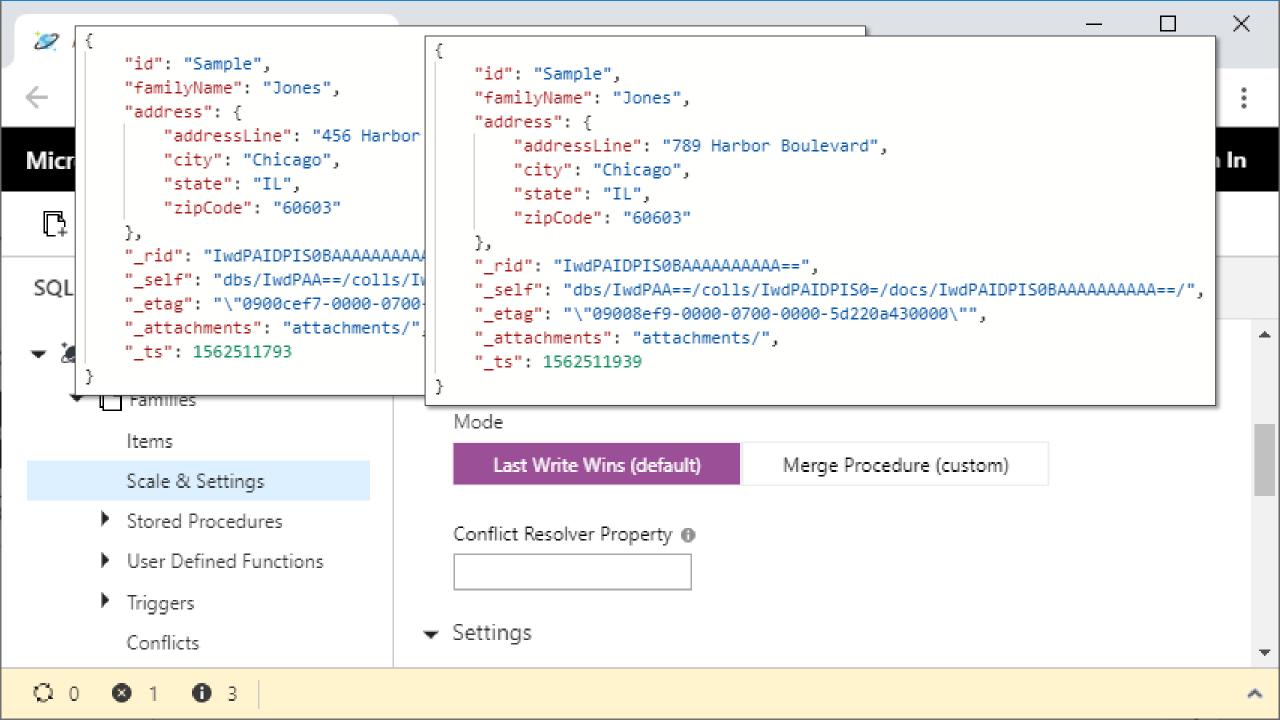


## Multi-master Conflict Resolution

#### Every region is enabled for writes, inviting conflicts

Three options for conflict resolution





## Multi-master Conflict Resolution

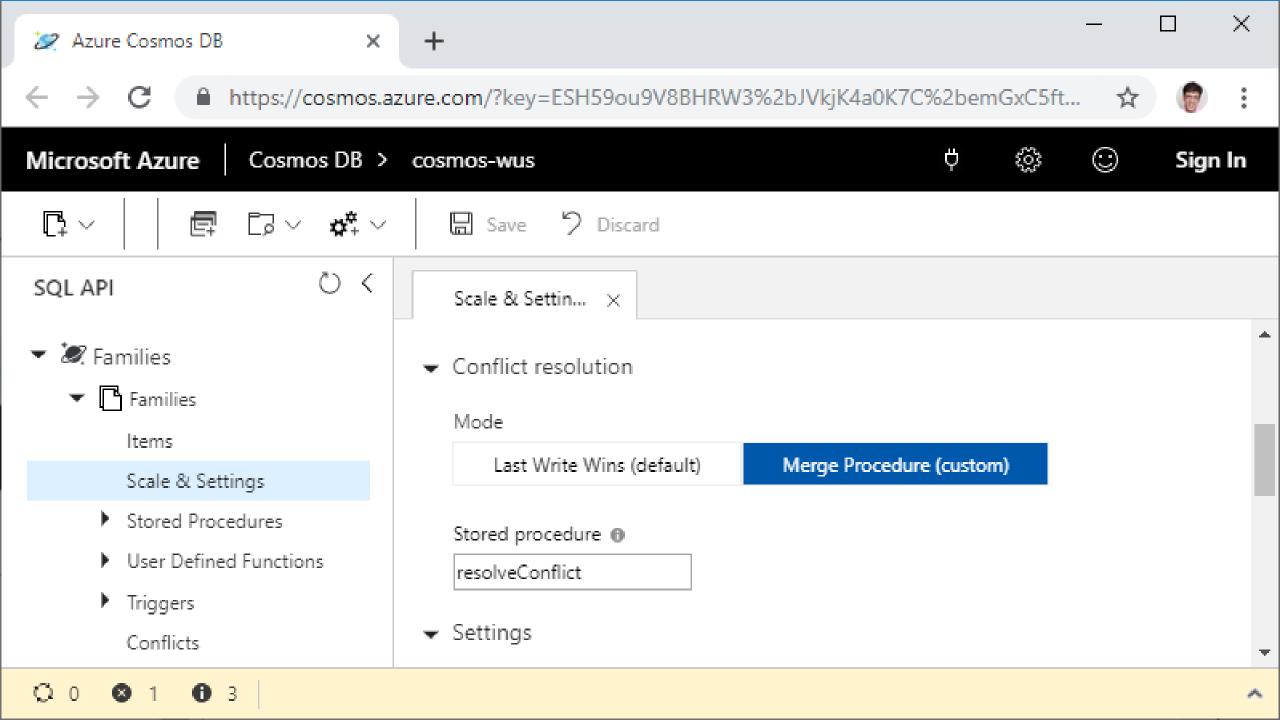
#### **Every region is enabled for writes**

Three options for conflict resolution

#### Last writer

Based on highest \_ts (or any other numeric property)





## Multi-master Conflict Resolution

#### **Every region is enabled for writes**

Three options for conflict resolution

#### Last writer

Based on highest \_ts (or any other numeric property)

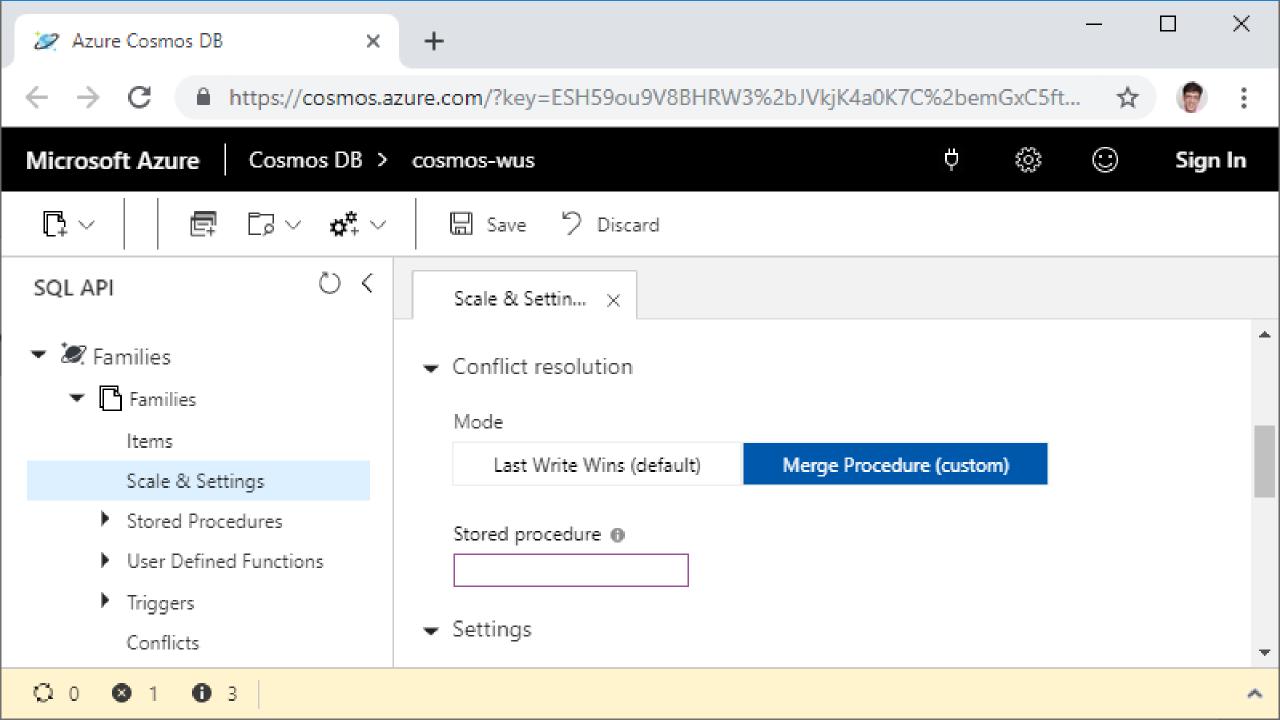
#### **Custom procedure**

Based on stored procedure result

#### **Conflict feed**

Offline resolution





## Replication and Pricing

#### No additional cost

No charge for the replication service

### **Egress charges**

Charged for outbound traffic during replication

# Container cost per region

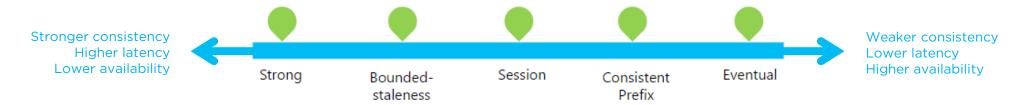
Throughput and storage charged separately in each region



## Replication and Consistency

#### How do you ensure consistent reads across replicas?

Define a consistency level



### Replication within a region

 Data moves extremely fast (typically, within 1ms) between neighboring racks

### Global replication

- It takes hundreds of milliseconds to move data across continents



## Five Consistency Levels

Strong

No dirty reads

#### **Bounded staleness**

Dirty reads possible

Bounded by time and updates

#### Session

No dirty reads for writers (read your own writes)

Dirty reads possible for other users

#### **Consistent prefix**

Dirty reads possible

Reads never see out-of-order writes

#### **Eventual**

Stale reads possible No guaranteed order

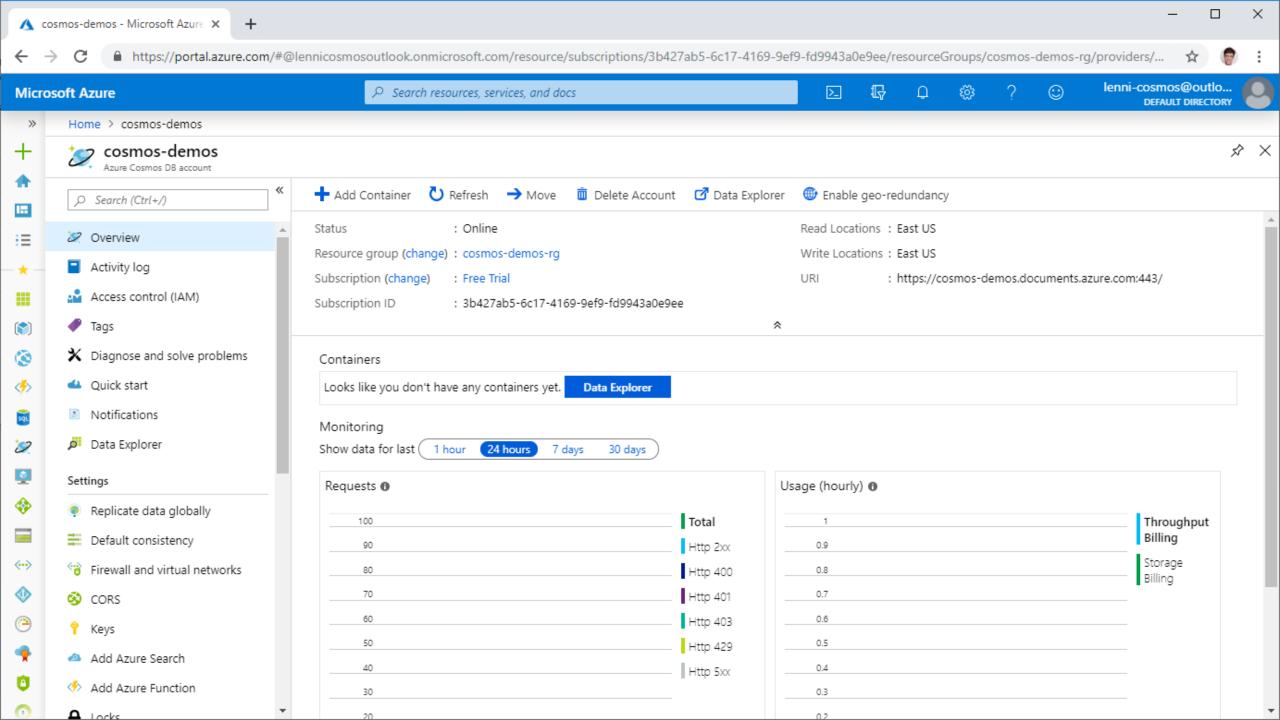


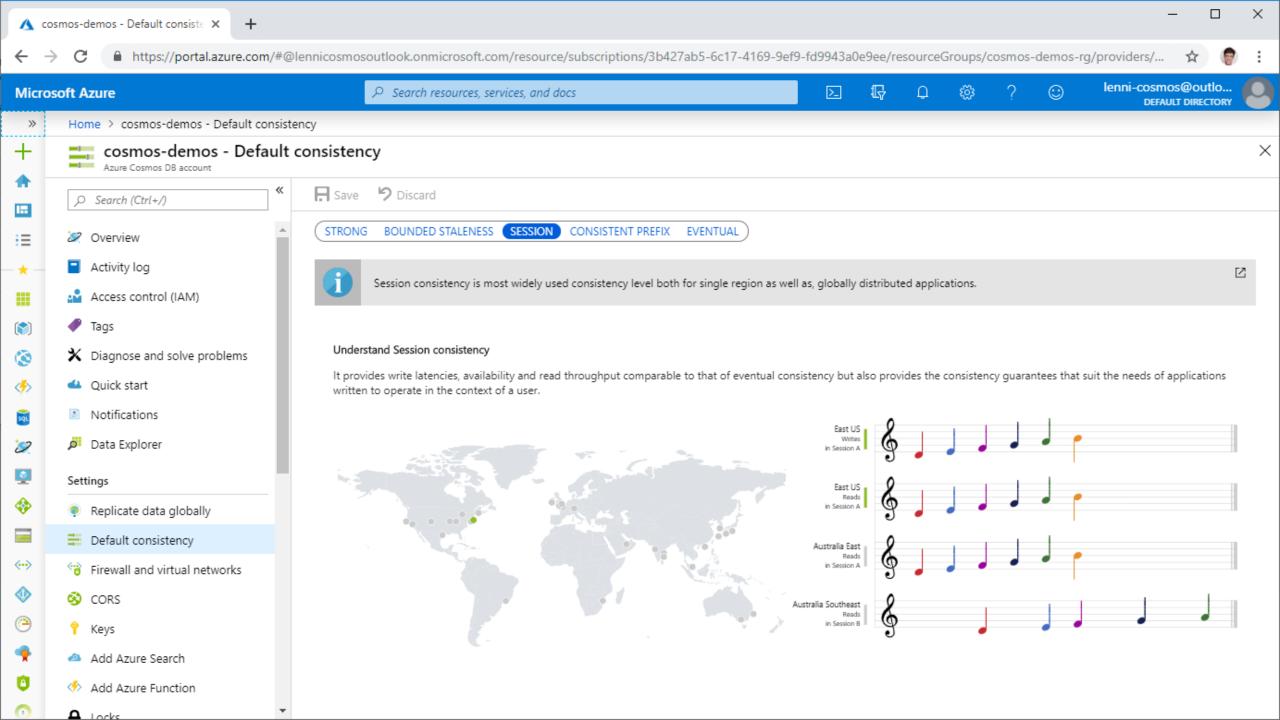
# Setting the Consistency Level

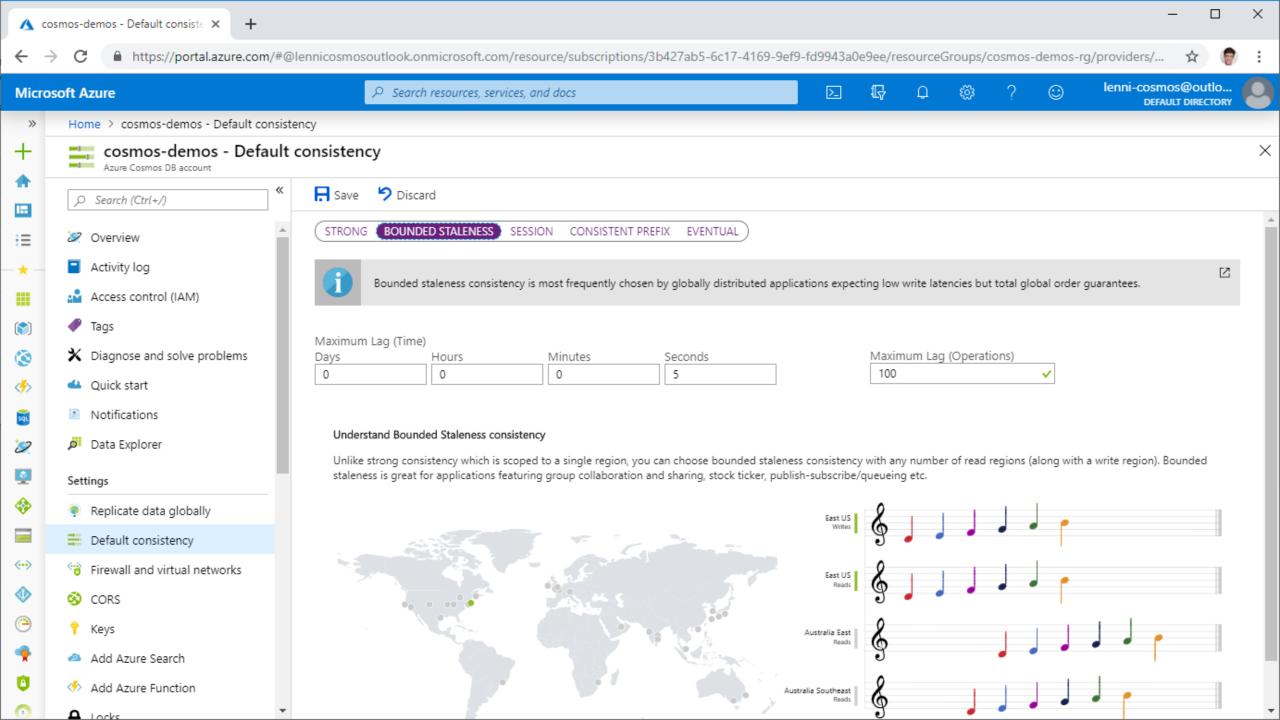
# Set default for entire account

Can be changed at any time









## Setting the Consistency Level

# Set default for entire account

Can be changed at any time

# Override at the request level

Any request can weaken the default consistency level

new ItemRequestOptions { ConsistencyLevel = ConsistencyLevel. }

- BoundedStaleness
- ConsistentPrefix
- Eventual
- Session



## Summary



#### Global distribution

- Point and click geo-replication
- Multi-master

#### **Preferred regions**

- Automatic failover

### Five consistency levels

- Strong
- Bounded staleness
- Session
- Consistent prefix
- Eventual

