2. Develop for Azure Storage -> 2.1 Develop solutions that use Azure Cosmos DB -> 2.1.3 Implement change feed notifications

1. What is the change feed in Azure Cosmos DB?
2. What types of changes does the change feed capture?
3. How do you read from the change feed using the SDK?
4. What is the difference between manual polling vs. Change Feed Processor?
5. How do you implement the Change Feed Processor in .NET?
6. How do you scale out a change feed listener?
7. What are common use cases for the change feed?
8. What is lease container and why is it required?
9. How do you resume reading from a specific point in the change feed?
10. What are best practices for change feed implementations?

---

1. What is the change feed in Azure Cosmos DB?

- A persistent, ordered log of item changes (inserts and updates) in a container.
- Enables event-driven processing without polling the whole dataset.

---

2. What types of changes does the change feed capture?

- **Creates and updates only.**
- **Deletes are not included.** You must implement soft delete patterns if needed.

---

3. How do you read from the change feed using the SDK?

```
var iterator = container.GetChangeFeedIterator<MyItem>(
   ChangeFeedStartFrom.Beginning(), ChangeFeedMode.Incremental);
while (iterator.HasMoreResults)
{
   var response = await iterator.ReadNextAsync();
   foreach (var item in response)
   {
      // Process item
   }
}
```

---

4. What is the difference between manual polling vs. Change Feed Processor?

- **Manual polling**: Directly queries the feed; full control but must manage state and scaling.
- **Change Feed Processor**: Auto-scales and handles lease/state tracking via a lease container.

---

5. How do you implement the Change Feed Processor in .NET?

```
var processor = container
   .GetChangeFeedProcessorBuilder<MyItem>("myProcessor", async (changes, token) =>
   {
      foreach (var item in changes) { /* process */ }
   })
   .WithInstanceName("worker1")
   .WithLeaseContainer(leaseContainer)
   .Build();
await processor.StartAsync();
```

- Requires a **lease container** for tracking progress.

6. How do you scale out a change feed listener?

- Use **multiple instances** of Change Feed Processor with the same lease container.
- The processor **automatically partitions** work across instances.

---

7. What are common use cases for the change feed?

- Event-driven processing (e.g., send emails, process orders)
- Real-time analytics
- Data movement to other stores (e.g., SQL, Blob Storage)
- Cache invalidation or sync

---

8. What is lease container and why is it required?

- A **separate Cosmos DB container** used by Change Feed Processor to track progress.
- Stores checkpoints and ownership info for scaling and fault-tolerance.

---

9. How do you resume reading from a specific point in the change feed?

- **Change Feed Processor** resumes automatically via lease container.
- Manual method: use ChangeFeedStartFrom.Time() or ChangeFeedStartFrom.ContinuationToken().

---

10. What are best practices for change feed implementations?

- Use **dedicated lease container** in same database.
- Ensure **idempotent processing logic**.
- Handle **throttling and retries** using SDK's retry policies.
- Monitor **lag and exceptions** for performance tuning.