Enterprise Computing Solutions - Education Services

TRAINING OFFERING

Sie erreichen uns hier

Freistädterstraße 236, A-4040 Linz

Email: education.ecs.at@arrow.com
Phone: +43 1 370 94 40 - 34

DP-420T00: Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB

CODE:	LÄNGE:	PREIS:
MCS_DP-420T00	32 Hours (4 Tage)	€2,120.00

Description

In diesem Kurs lernen Entwickler, wie sie Anwendungen mit der SQL-API und dem SDK für Azure Cosmos DB erstellen. Die Teilnehmer lernen, wie man effiziente Abfragen schreibt, Indizierungsrichtlinien erstellt, Ressourcen verwaltet und bereitstellt und allgemeine Operationen mit dem SDK durchführt.

Lernziel

Erstellung und Konfiguration von Azure Cosmos DB SQL API-Konto, Datenbank und Container
Verwendung des .NET SDK zur Verwaltung von Ressourcen und Durchführung von Operationen
Ausführung von Abfragen unterschiedlicher Komplexität
Entwerfung einer Datenmodellierungs- und Partitionierungsstrategie
Optimierung von Abfragen und Indizes basierend auf den Eigenschaften einer Anwendung
Verwendung des Azure Resource Manager zur Verwaltung von Konten und Ressourcen mit CLI oder JSON und Bicep-Vorlagen

Zielgruppe

Dieses Seminar richtet sich an:
Software-Ingenieure, die mit der Erstellung von Cloud-nativen Lösungen betraut sind, die die Azure Cosmos DB SQL API und ihre verschiedenen SDKs nutzen.
Sie sind vertraut mit C#, Python, Java oder JavaScript.
Sie haben außerdem Erfahrung im Schreiben von Code, der mit einer SQL- oder NoSQL-Datenbankplattform interagiert.

Voraussetzungen

Für dieses Seminar werden folgende Vorkenntnisse empfohlen:
Kenntnisse über Microsoft Azure und die Fähigkeit, sich im Azure-Portal zurechtzufinden (entspricht AZ-900)
Erfahrung im Schreiben in einer von Azure unterstützten Sprache auf mittlerem Niveau. (C#, JavaScript, Python oder Java)
Fähigkeit, Code zu schreiben, der eine Verbindung zu einem SQL- oder NoSQL-Datenbankprodukt herstellt und Operationen darauf ausführt. (SQL Server, Oracle, MongoDB, Cassandra oder ähnlich)

Inhalt

Module 1: Get started with Azure Cosmos DB SQL API
Modern apps thrive on real-time data from different sources and shaped in different forms. These apps require a modern database that can handle the variety and velocity of data that will be thrown at it. In this module, we will explore Azure Cosmos DB and how the SQL API can solve some of the problems presented by modern applications.
Lessons
Introduction to Azure Cosmos DB SQL API
Try Azure Cosmos DB SQL API Lab : Exercise: Create an Azure Cosmos DB SQL API account
After completing this module, students will be able to:
Evaluate whether Azure Cosmos DB SQL API is the right database for your application
Describe how the features of the Azure Cosmos DB SQL API are appropriate for modern applications
Create a new Azure Cosmos DB SQL API account
Create database, container, and item resources for an Azure Cosmos DB SQL API account

Module 2: Plan and implement Azure Cosmos DB SQL API
Creating a new Azure Cosmos DB account often requires making a lot of configuration choices that can, at first, be daunting. While the defaults fit a lot of scenarios, it makes the most sense to familiarize yourself with the configuration options to ensure that your account and resources are optimally configured for your solution. In this module, you will learn how to prepare and configure an Azure Cosmos DB account and resources for a new solution.
Lessons
Plan Resource Requirements
Configure Azure Cosmos DB SQL API database and containers
Moving data into and out of Azure Cosmos DB SQL API
Lab : Exercise: Configure throughput for Azure Cosmos DB SQL API with the Azure portal
Lab : Exercise: Migrate existing data using Azure Data Factory
After completing this module, students will be able to:
- Evaluate various requirements of your application
- Plan for scale and retention requirements
- Configure throughput allocation
- Configure time-to-live values
- Migrate data using Azure services
- Migrate data using Spark or Kafka

Module 3: Connect to Azure Cosmos DB SQL API with the SDK
There are various SDKs available to connect to the Azure Cosmos DB SQL API from many popular programming languages including, but not limited to .NET (C#), Java, Python, and JavaScript (Node.js). In this module, you will get hands-on with with the .NET SDK for the Azure Cosmos DB SQL API.
Lessons
Use the Azure Cosmos DB SQL API SDK
Configure the Azure Cosmos DB SQL API SDK
Lab : Exercise: Configure the Azure Cosmos DB SQL API SDK for offline development
Lab : Exercise: Connect to Azure Cosmos DB SQL API with the SDK
After completing this module, students will be able to:
- Integrate the Microsoft.Azure.Cosmos SDK library from NuGet
- Connect to an Azure Cosmos DB SQL API account using the SDK and .NET
- Configure the SDK for offline development
- Troubleshoot common connection errors
- Implement parallelism in the SDK
- Configure logging using the SDK

Module 4: Access and manage data with the Azure Cosmos DB SQL API SDKs
The SQL API SDK for Azure Cosmos DB is used to perform various point operations, perform transactions, and to process bulk data. In this module, you will use the SDK to manipulate documents either individually or in groups.
Lessons
Implement Azure Cosmos DB SQL API point operations
Perform cross-document transactional operations with the Azure Cosmos DB SQL API
Process bulk data in Azure Cosmos DB SQL API
Lab : Exercise: Create and update documents with the Azure Cosmos DB SQL API SDK
Lab : Exercise: Batch multiple point operations together with the Azure Cosmos DB SQL API SDK
Lab : Exercise: Move multiple documents in bulk with the Azure Cosmos DB SQL API SDK
After completing this module, students will be able to:
- Perform CRUD operations using the SDK
- Configure TTL for a specific document
- Implement optimistic concurrency control for an operation
- Create a transactional batch and review results
- Create a bulk operation
- Review the results of a bulk operation
- Implement bulk operation best practices

Module 5: Execute queries in Azure Cosmos DB SQL API
The Azure Cosmos DB SQL API supports Structured Query Language (SQL) as a JSON query language. In this module, you will learn how to create efficient queries using the SQL query language.
Lessons
Query the Azure Cosmos DB SQL API
Author complex queries with the Azure Cosmos DB SQL API
Lab : Exercise: Paginate cross-product query results with the Azure Cosmos DB SQL API SDK
Lab : Exercise: Execute a query with the Azure Cosmos DB SQL API SDK
After completing this module, students will be able to:
- Create and execute a SQL query
- Project query results
- Use built-in functions in a query
- Implement a corelated subquery
- Create a cross-product query

Module 6: Define and implement an indexing strategy for Azure Cosmos DB SQL API

By default, Azure Cosmos DB automatically indexes all paths of documents stored using the SQL API. This is great for developing new applications as you can create complex queries almost immediately. As your application matures, you can customize your indexing policy to better match the needs of your solution. In this module, you will learn how to create a custom indexing policy.

Lessons

Define indexes in Azure Cosmos DB SQL API

Customize indexes in Azure Cosmos DB SQL API

Lab : Exercise: Review the default index policy for an Azure Cosmos DB SQL API container with the portal

Lab : Exercise: Configure an Azure Cosmos DB SQL API container's index policy with the portal

After completing this module, students will be able to:

View and understand the default indexing policy for a SQL API container

Customize the indexing policy for a container

Use a composite index in an indexing policy

Module 7: Integrate Azure Cosmos DB SQL API with Azure services

Azure Cosmos DB has tight integration available with many other Azure servicers such as Azure Functions, Azure Cognitive Search, Azure Event Hubs, Azure Storage, Azure Data Factory, and Azure Stream Analytics. Going even further, you can use the change feed to integrate Azure Cosmos DB with many other services both in and out of Azure. In this module, we will integrate Azure Cosmos DB with both Azure Functions and Azure Cognitive Search. We will also explore the change feed using the SDK.

Lessons

Consume an Azure Cosmos DB SQL API change feed using the SDK

Handle events with Azure Functions and Azure Cosmos DB SQL API change feed

Search Azure Cosmos DB SQL API data with Azure Cognitive Search

Lab : Exercise: Archive Azure Cosmos DB SQL API data using Azure Functions

Lab : Exercise: Process change feed events using the Azure Cosmos DB SQL API SDK

Lab : Exercise: Archive data using Azure Functions and Azure Cosmos DB SQL API

After completing this module, students will be able to:

Process change feed events using the SDK

Implement change feed best practices

Create an Azure Functions trigger for Azure Cosmos DB

Create an Azure Functions input for Azure Cosmos DB

Index Azure Cosmos DB data in Azure Cognitive Search

Module 8: Implement a data modeling and partitioning strategy for Azure Cosmos DB SQL API

Azure Cosmos DB is both horizontally scalable and nonrelational. To achieve this level of scalability, users need to understand the concepts, techniques, and technologies unique to NoSQL databases for modeling and partitioning data. In this module, you will model and partition data appropriately for a NoSQL database such as Azure Cosmos DB SQL API.

Lessons

Model and partition your data in Azure Cosmos DB

Optimize databases by using advanced modeling patterns for Azure Cosmos DB

Lab : Exercise: Measure performance for customer entities

Lab : Exercise: Advanced modeling patterns

After completing this module, students will be able to:

Identify application access patterns for an existing application

Decide when to embed or reference data

Use change feed to manage referential integrity

Combine multiple entities in a single container

Denormalize aggregated data in a single container

Module 9: Design and implement a replication strategy for Azure Cosmos DB SQL API

Today's applications are required to be highly responsive and always online. To achieve low latency and high availability, instances of these applications need to be deployed in datacenters that are close to their users. In this module, you will explore how to replicate data and manage consistency across the globe using Azure Cosmos DB SQL API.

Lessons

Configure replication and manage failovers in Azure Cosmos DB

Use consistency models in Azure Cosmos DB SQL API

Configure multi-region write in Azure Cosmos DB SQL API

Lab : Exercise: Configure consistency models in the portal and the Azure Cosmos DB SQL API SDK

Lab : Exercise: Connect to different regions with the Azure Cosmos DB SQL API SDK

Lab : Exercise: Connect to a multi-region write account with the Azure Cosmos DB SQL API SDK

After completing this module, students will be able to:

Distribute data across various geographies

Define automatic failover policies

Perform manual failovers

Configure default consistency model

Change per-session consistency model

Configure multi-region write in the SDK

Create a custom conflict resolution policy

Module 10: Optimize query performance in Azure Cosmos DB SQL API

Azure Cosmos DB offers a rich set of database operations that operate on the items within a container. The cost associated with each of these operations varies based on the CPU, IO, and memory required to complete the operation. In this module, you will explore how to manage indexing policies and edit queries to minimize per-query request unit (RU) cost.

Choosing indexes in Azure Cosmos DB SQL API
Optimize queries in Azure Cosmos DB SQL API
Lessons Implement integrated cache

Lab : Exercise: Optimize an Azure Cosmos DB SQL API container's index policy for common operations
Lab : Exercise: Optimize an Azure Cosmos DB SQL API container's index policy for a specific query
Review and compare read-heavy vs. write-heavy index patterns
Update indexing policy to optimize index performance
Measure cost of a query in request units (RUs)
Measure cost of point operations
Work with item and query integrated cache
After completing this module, students will be able to: Configure integrated cache staleness

Module 11: Administrating and Monitoring tasks for an Azure Cosmos DB SQL API solution
When you have critical applications and business processes relying on Azure resources such as Azure Cosmos DB, you want to monitor those resources for their availability, performance, and operation. In this module, you will explore how to monitor events and performance of an Azure Cosmos DB account. You will also learn how to implement common security measures along with backup and restore in Azure Cosmos DB.
Lessons
Measure performance in Azure Cosmos DB SQL API
Monitor responses and events in Azure Cosmos DB SQL API
Implementing backup and restore for Azure Cosmos DB SQL API
Implement security in Azure Cosmos DB SQL API
Lab : Exercise: Troubleshoot an application using the Azure Cosmos DB SQL API SDK
Lab : Exercise: Use Azure Monitor to analyze an Azure Cosmos DB SQL API account
Lab : Exercise: Recover a database or container from a recovery point
Lab : Exercise: Store Azure Cosmos DB SQL API account keys in Azure Key Vault
Observe rate-limiting events in a container or database
Query resource logs using Azure Monitor
Review and observe transient and rate-limiting errors
Configure alerts
Configure continuous backup and recovery
Perform a point-in-time recovery
Use role-based access control (RBAC)
After completing this module, students will be able to: Access account resources using Azure AD and Microsoft Identity PIatform
Module 12: Manage an Azure Cosmos DB SQL API solution using DevOps practices
Once an Azure Cosmos DB SQL API account is ready to go through a release lifecycle, it's not uncommon for an operations team to attempt to automate the creation of Azure Cosmos DB resources in the cloud. Automation makes it easier to deploy new environments, restore past environments, or scale a service out. In this module, you will explore how to use Azure Resource Manager to manage an Azure Cosmos DB account and its child resources using JSON templates, Bicep templates, or the Azure CLI.
Lessons
Write scripts for Azure Cosmos DB SQL API
Create resource template for Azure Cosmos DB SQL API
Lab : Exercise: Adjust provisioned throughput using an Azure CLI script
Lab : Exercise: Create an Azure Cosmos DB SQL API container using Azure Resource Manager templates
After completing this module, students will be able to:
View arguments, groups, and subgroups for a specific CLI command
Create Azure Cosmos DB accounts, databases, and containers using the CLI
Manage an indexing policy using the CLI
Configure database or container throughput using the CLI
Initiate failovers and manage failover regions using the CLI
Identify the three most common resource types for Azure Cosmos DB SQL API accounts
Create and deploy a JSON Azure Resource Manager template for Azure Cosmos DB SQL API
Create and deploy a Bicep Azure Resource Manager template for Azure Cosmos DB SQL API
Manage throughput and index policies using JSON or Bicep templates

Module 13: Create server-side programming constructs in Azure Cosmos DB SQL API
Azure Cosmos DB provides language-integrated, transactional execution of JavaScript. When using the SQL API in Azure Cosmos DB, you can write stored procedures, triggers, and user-defined functions (UDFs) in the JavaScript language. In this module, you will author JavaScript logic that executes directly inside the database engine.
Lessons
Build multi-item transactions with the Azure Cosmos DB SQL API
Expand query and transaction functionality in Azure Cosmos DB SQL API
Lab : Exercise: Implement and then use a UDF using the SDK
Lab : Exercise: Create a stored procedure with the Azure Portal After completing this module, students will be able to:

Author stored procedure
Rollback stored procedure transaction
Create UDF
Create pre-* and post-* triggers

Kurstermine

Auf Anfrage. Bitte kontaktieren Sie uns

Zusätzliche Information

Diese Schulung ist auch als Vor-Ort-Schulung verfügbar. Bitte kontaktieren Sie uns, um mehr zu erfahren.