

Monitor and troubleshoot an Azure Cosmos DB for NoSQL solution



Agenda



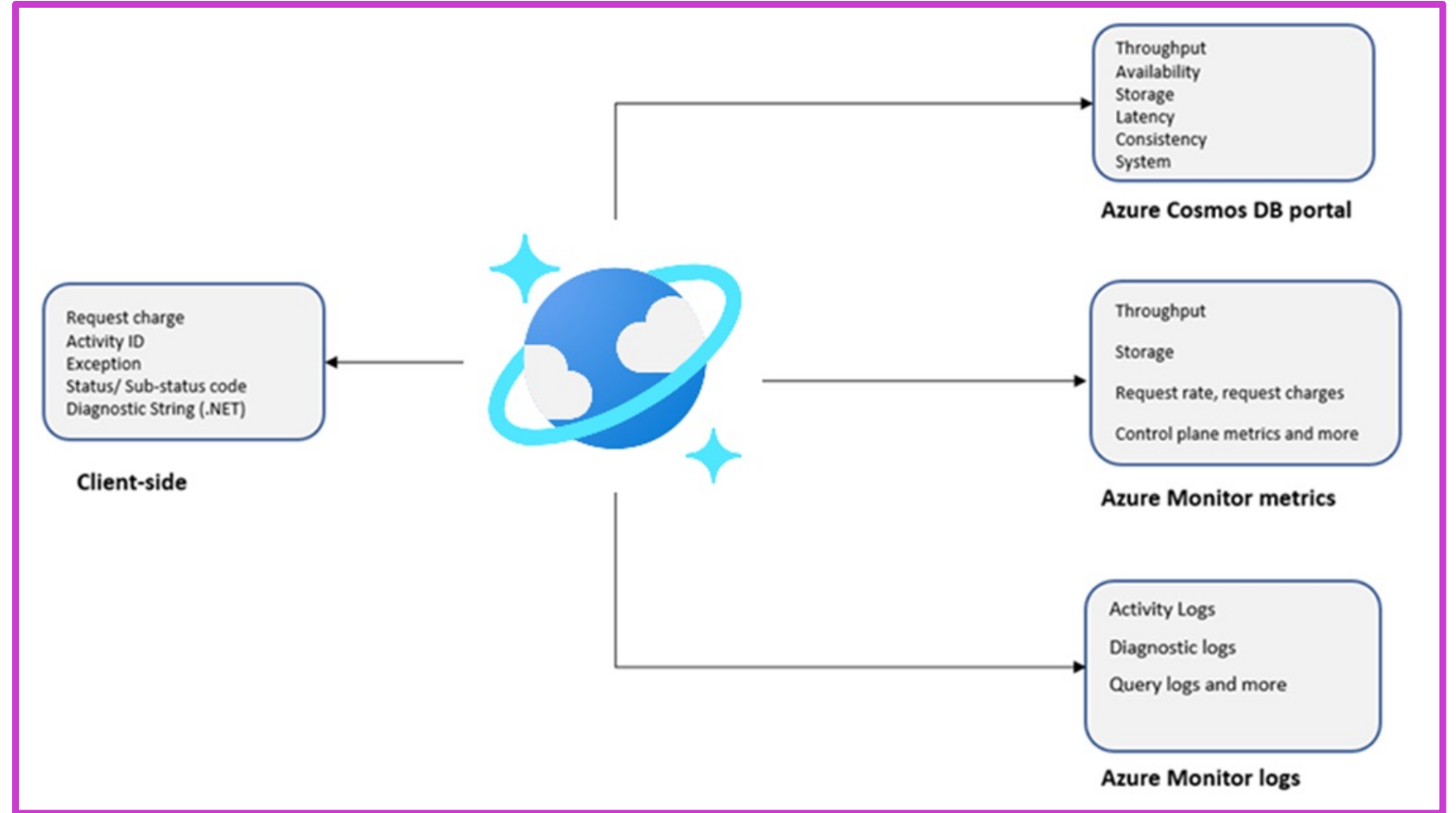
- Measure performance in Azure Cosmos DB for NoSQL
- Monitor responses and events in Azure Cosmos DB for NoSQL
- Implementing backup and restore for Azure Cosmos DB for NoSQL
- Implement security in Azure Cosmos DB for NoSQL

Measure performance in Azure Cosmos DB for NoSQL



Understand Azure Monitor

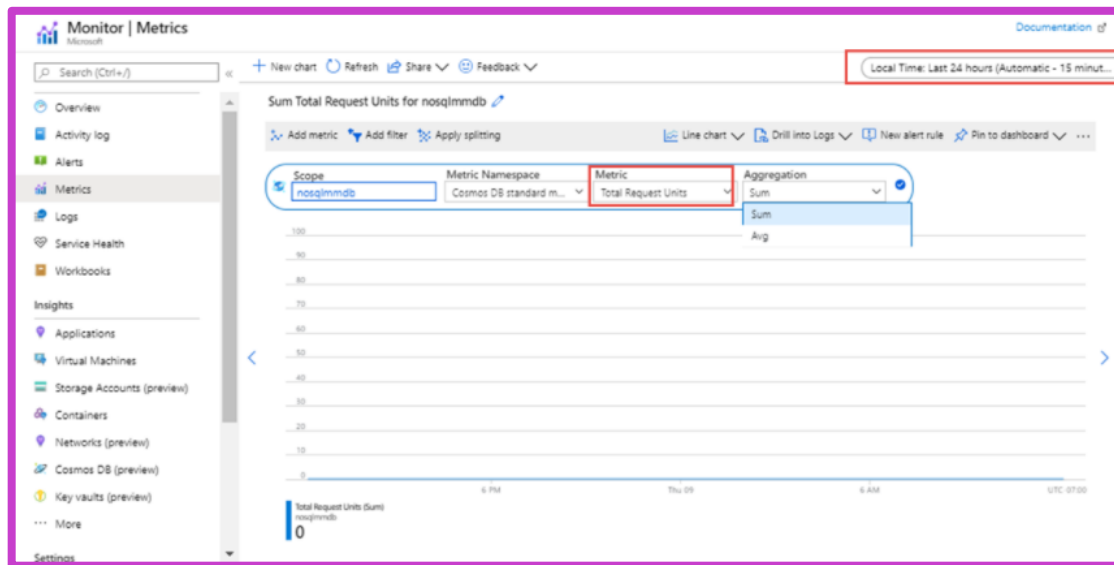
Azure Monitor is used to monitor the Azure resource availability, performance, and operations metrics.



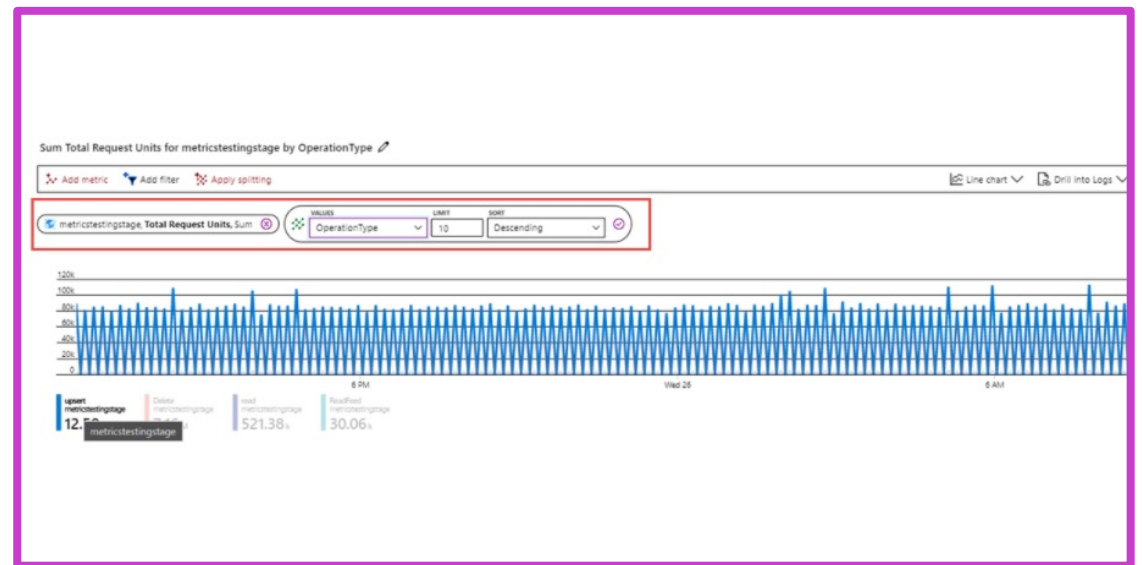
Measure throughput

The Total Request Units metric can then be used to analyze those operations with the highest throughput.

View the Total Request Unit metrics



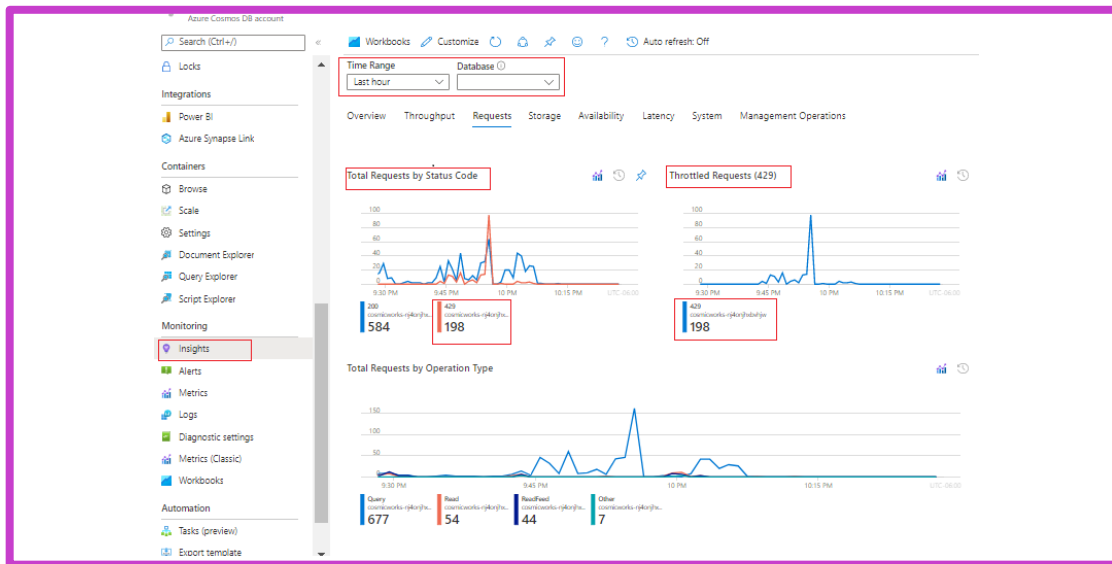
Filter the Total Request Units further



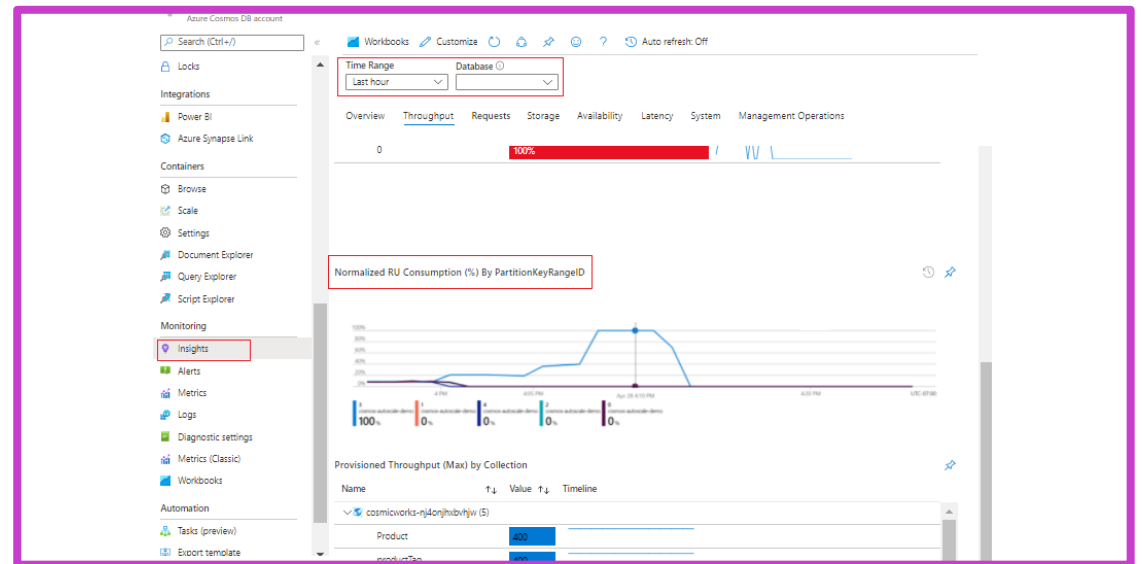
Observe rate-limiting events

The *429-status code* indicates that a *Request rate too large exception* has occurred. This exception means that Azure Cosmos DB requests are being rate limited. *

Review the Insights-Request charts for 429s



Review the Insights-Request charts for hot partitions



* Rate limiting exceptions can also be due to metadata request or transit service errors.

Query logs

Diagnostics settings are used to collect Azure Diagnostic Logs produced by Azure resources. These logs provide detailed resource operational data.

Create Azure Cosmos DB diagnostics settings

The screenshot shows the 'Diagnostic setting' configuration page in the Azure portal. The 'Diagnostic setting name' is 'DP420 Cosmos DB diagnostic logs'. Under the 'Logs' section, 'Category groups' are 'audit' and 'allLogs'. The 'Categories' list includes 'DataPlaneRequests', 'MongoRequests', 'QueryRuntimeStatistics', 'PartitionKeyStatistics', 'PartitionKeyRUConsumption', 'ControlPlaneRequests', 'CassandraRequests', 'GremlinRequests', and 'TableApiRequests'. The 'Metrics' section has 'Requests'. Under 'Destination details', 'Send to Log Analytics workspace' is checked. The 'Subscription' is 'wwl420labs' and the 'Log Analytics workspace' is 'wwl420labslaws (westus)'. The 'Destination table' is 'Azure diagnostics' with a 'Resource specific' link. Other options like 'Archive to a storage account', 'Stream to an event hub', and 'Send to partner solution' are unchecked.

Troubleshoot issues with KQL diagnostics queries

```
// AzureDiagnostics queries
AzureDiagnostics
| where TimeGenerated >= ago(1h)
| where ResourceProvider=="MICROSOFT.DOCUMENTDB" and
Category=="DataPlaneRequests"
| summarize OperationCount = count(),
TotalRequestCharged=sum(todouble(requestCharge_s)) by
OperationName
| order by TotalRequestCharged desc
```

```
// Resource-specific Queries
CDBDataPlaneRequests
| where TimeGenerated >= ago(1h)
| summarize OperationCount = count(),
TotalRequestCharged=sum(todouble(RequestCharge)) by
OperationName
| order by TotalRequestCharged desc
```


Lab – Use Azure Monitor to analyze an Azure Cosmos DB for NoSQL account



- Prepare your development environment
- Create an Azure Cosmos DB for NoSQL account
- Import the Microsoft.Azure.Cosmos and Newtonsoft.Json libraries into a .NET script
- Run a script to create the containers and the workload
- Use Azure Monitor to Analyze the Azure Cosmos DB account usage

Monitor responses and events in Azure Cosmos DB for NoSQL



Review common response codes

Azure Cosmos DB for NoSQL operations that create, query, or manage container documents, will return an HTTP operation status code.

Status Code	Name
200	OK
201	Created
204	No Content
304	Not Modified
400	Bad Request
403	Forbidden
404	Not Found
408	Request timeout
409	Conflict
413	Entity Too Large
429	Too many requests
500	Internal Server Error
503	Service Unavailable

Understand transient errors

We can identify and troubleshoot Azure Cosmos DB service unavailable exceptions when our request returns status code 503.

Required ports are blocked

Connection mode	Supported protocol	API/Service port
Gateway	HTTPS	SQL (443)
Direct	TCP	When using public/service endpoints: ports in the 10000 through 20000 range. When using private endpoints: ports in the 0 through 65535 range

Client-side transient connectivity issues

`TransportException: A client transport error occurred: The request timed out while waiting for a server response.`

`(Time: xxx, activity ID: xxx, error code: ReceiveTimeout [0x0010], base error: HRESULT 0x80131500`

Service Outage

Check the *Azure status page* to see if there's an ongoing issue.

Review rate limiting errors

Requests return status code 429 for the exception request rate too large status code, indicating that your requests against Azure Cosmos DB are being rate-limited.

KQL query to determine which request types are causing 429 exceptions

```
// Resource-specific Queries
AzureDiagnostics
| where TimeGenerated >= ago(24h)
| where Category == "DataPlaneRequests"
| summarize throttledOperations = dcountif(activityId_g, statusCode_s == 429), totalOperations = dcount(activityId_g),
totalConsumedRUPerMinute = sum(todouble(requestCharge_s)) by databaseName_s, collectionName_s, OperationName,
requestResourceType_s, bin(TimeGenerated, 1min) | extend averageRUPerOperation = 1.0 * totalConsumedRUPerMinute /
totalOperations | extend fractionOf429s = 1.0 * throttledOperations / totalOperations
| order by fractionOf429s desc
```

Rate-limiting on metadata requests

Review occurrences of 429 exception in the Azure Cosmos DB *Insight* report *Metadata Requests That Exceeded Capacity (429s)* under the *System* tab.

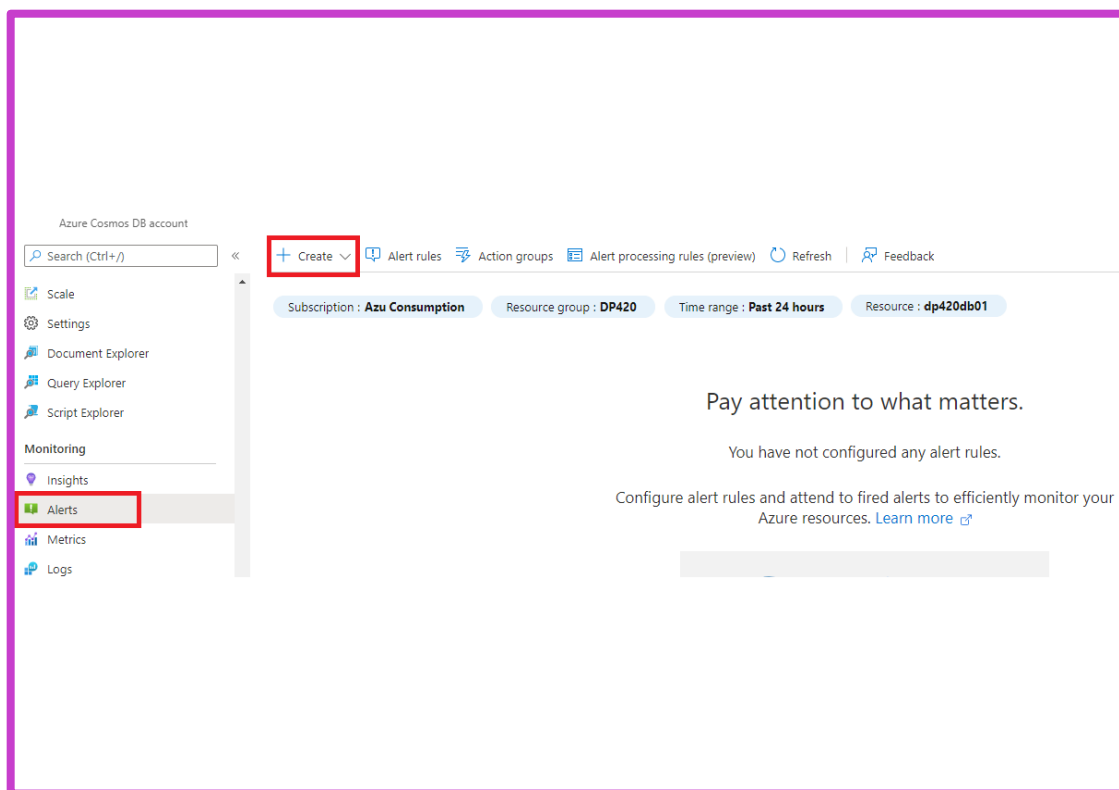
Rate-limiting due to transient service error

Retrying the request is the only recommended solution.

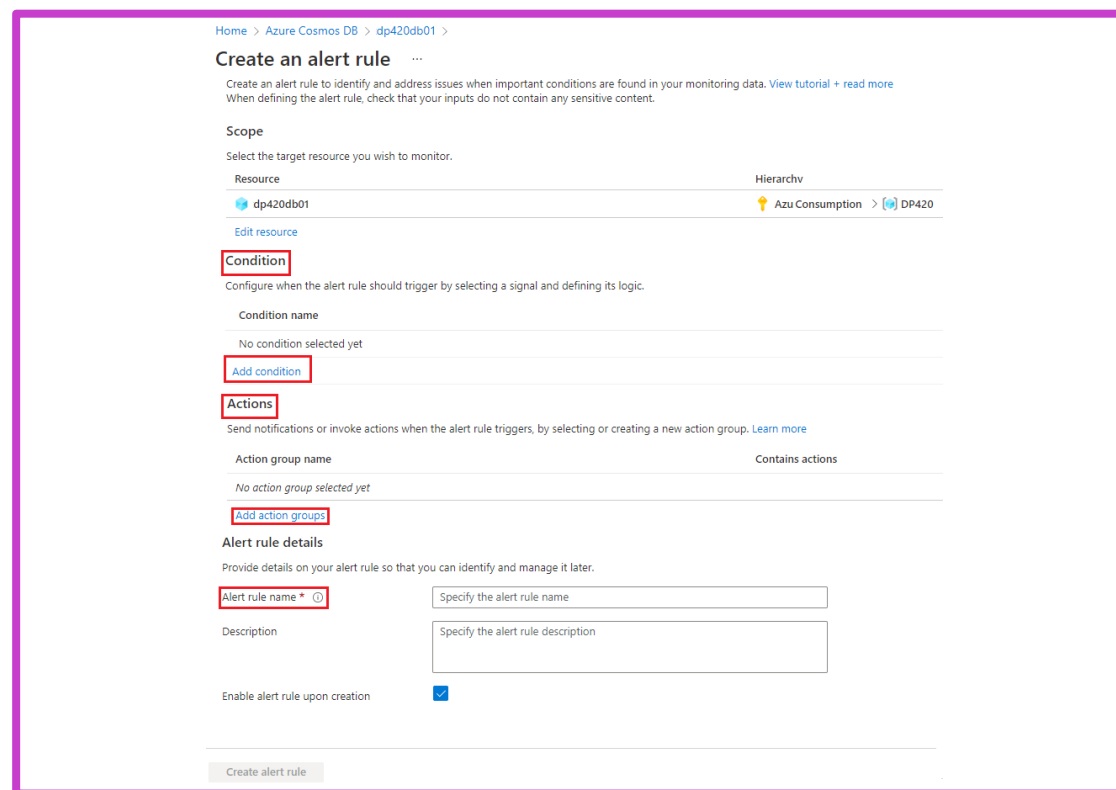
Configure Alerts

Azure Cosmos DB uses the Azure Monitor Service to set up and send alerts.

Create an alert



Create alert rules



Audit security

Azure Cosmos DB uses the Azure Monitor Service to set up and send alerts.

Activity Logs

Azure Cosmos DB account: **Cosmos DB Account**

Activity

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Cost Management

Quick start

Notifications

Data Explorer

Settings

Features

Replicate data globally

Default consistency

Backup & Restore

Firewall and virtual networks

Private Endpoint Connections

CORS

Dedicated Gateway

Keys

Advisor Recommendations

Advanced security (preview)

Looking for Log Analytics? In Log Analytics you can search for performance, diagnostics, health logs, and more. Visit Log Analytics

Search

Quick Insights

Subscription: Azure subscription 1

Event severity: All

Timespan: Last 6 hours

Resource group: DP420

Resource: student01cosmos01

Event category: All categories

Add Filter

13 items.

Operation name	Status	Time	Time stamp	Subscription	Event initiated by
List keys	Succeeded	5 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
Backup	Succeeded	5 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
List keys	Succeeded	7 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
Rotate keys	Succeeded	7 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
List keys	Succeeded	7 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
Backup	Succeeded	7 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
List keys	Succeeded	8 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
List keys	Succeeded	8 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
Get Connection Strings	Succeeded	8 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
Get Connection Strings	Succeeded	8 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
Read database account readonl	Succeeded	8 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
List keys	Succeeded	8 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...
Backup	Succeeded	8 minutes a...	Fri Nov 26 2...	Azure subscription 1	dp420student01@...

Azure Resource Logs

- Enable Azure resource logs for Cosmos DB.
- Enable the auditing control plane under Diagnostics settings.

Lab – Troubleshoot an application using the Azure Cosmos DB for NoSQL SDK



- Prepare your development environment
- Create an Azure Cosmos DB for NoSQL account
- Import the Microsoft.Azure.Cosmos library into a .NET script
- Run a script to create menu-driven options to insert and delete documents
- Time to insert and delete documents

Implementing backup and restore for Azure Cosmos DB for NoSQL



Evaluate periodic backup

Azure Cosmos DB takes automatic backups of your data at regular periodic intervals.

Backup Storage Redundancy	Change the default backup interval and retention period	To request to restore a backup	Consider restoring a backup when you ...	Costs of Extra backups	Manage your own backups
<ul style="list-style-type: none">• Geo-redundant• Zone-redundant• Locally redundant	<ul style="list-style-type: none">• Backup Interval• Backup Retention• Backup storage redundancy	Open a request ticket or call the Azure support team.	<ul style="list-style-type: none">• ... deleted the entire Azure Cosmos DB account.• ... deleted one or more Azure Cosmos DB databases.• ... deleted one or more Azure Cosmos DB containers.• ... deleted or modified the Azure Cosmos DB items within a container.	<ul style="list-style-type: none">• Two backups included with the account for free.• Extra backups will be charged on a region-based backup-storing pricing.	<ul style="list-style-type: none">• Azure Data Factory• Change feed

Configure continuous backup and recovery

When using the continuous backups mode, backups are continuously taken in every region where the Azure Cosmos DB account exists.

Backup Storage Redundancy	Change backup options	Continuous backup mode charges	Limitations – Not supported	Limitations
<ul style="list-style-type: none">• Locally redundant by default• Zone-redundant when using Availability zones	<ul style="list-style-type: none">• Only option is to enable Continuous Backups• Once set on a new or existing account can not be changed	<ul style="list-style-type: none">• Backup storage space.• Restore cost.• A separate charge will be added every time a restore is started.	<ul style="list-style-type: none">• Cosmos accounts using customer-managed keys.• Multi-region write accounts.• Accounts that create unique indexes after the container is created.	<ul style="list-style-type: none">• You can't restore an account into a region where the source account did not exist.• The retention period is 30 days and can't be changed.• Point in time restore always restores to a new Azure Cosmos DB account.

Perform a point-in-time recovery

Point-in-time recovery will allow you to choose any timestamp within the up to 30-days backup retention period and restore a combination of Azure DB containers, databases, or the accounts.

Restore Scenarios

- Restore deleted account
- Restore data of an account in a particular region
- Recover from an accidental write or delete operation within a container with a known restore timestamp
- Restore an account to a previous point in time before the accidental delete of the database
- Restore an account to a previous point in time before the accidental delete or modification of the container properties

Lab – Recover a database or container from a recovery point



- Create an Azure Cosmos DB for NoSQL account
- Add a database and two containers to the account
- Add a database and two containers to the account
- Change the default backup mode to continuous (Optional if feature not enabled during the account creation)
- Delete one the salesOrder documents
- Restore the database to the point before you deleted the salesOrder document
- Delete the customer container
- Restore the database to the point before you deleted the salesOrder document
- Review the data restored and cleanup

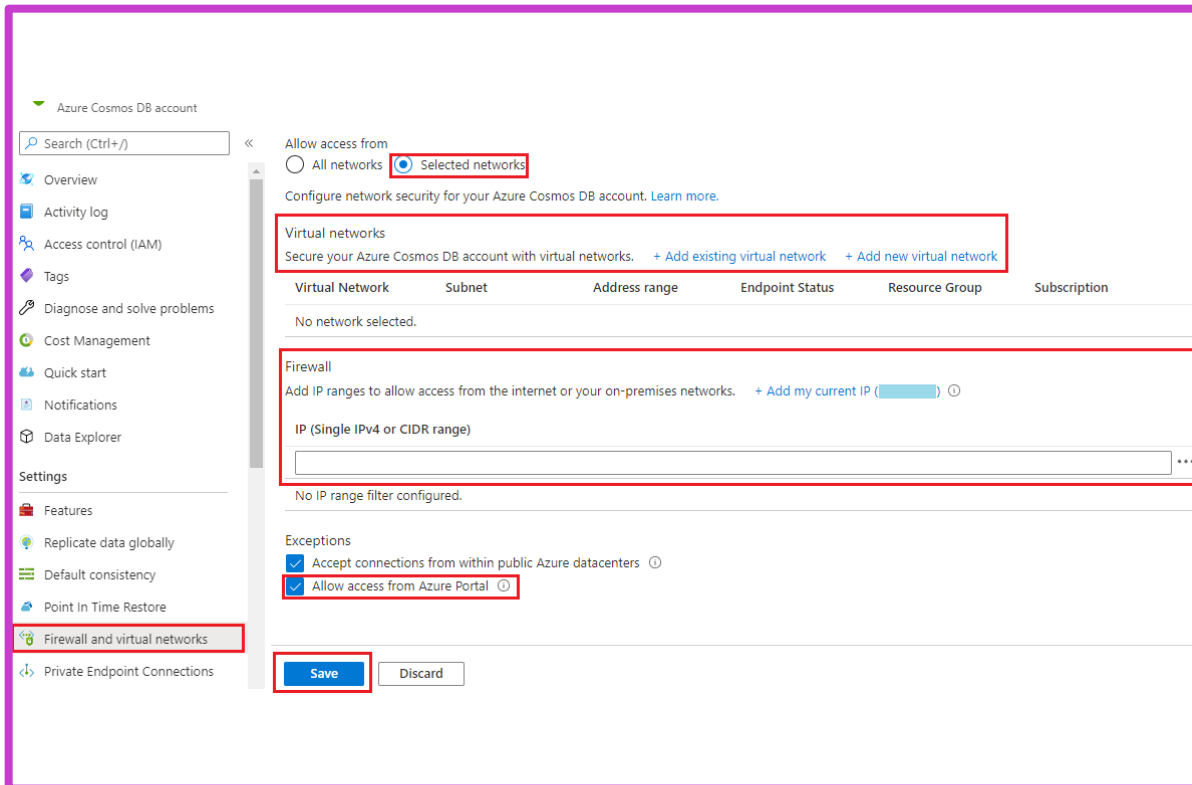
Implement security in Azure Cosmos DB for NoSQL



Implement network-level access control

Azure Cosmos DB supports IP-based access controls for inbound firewall support.

Configure an IP firewall by using the Azure portal



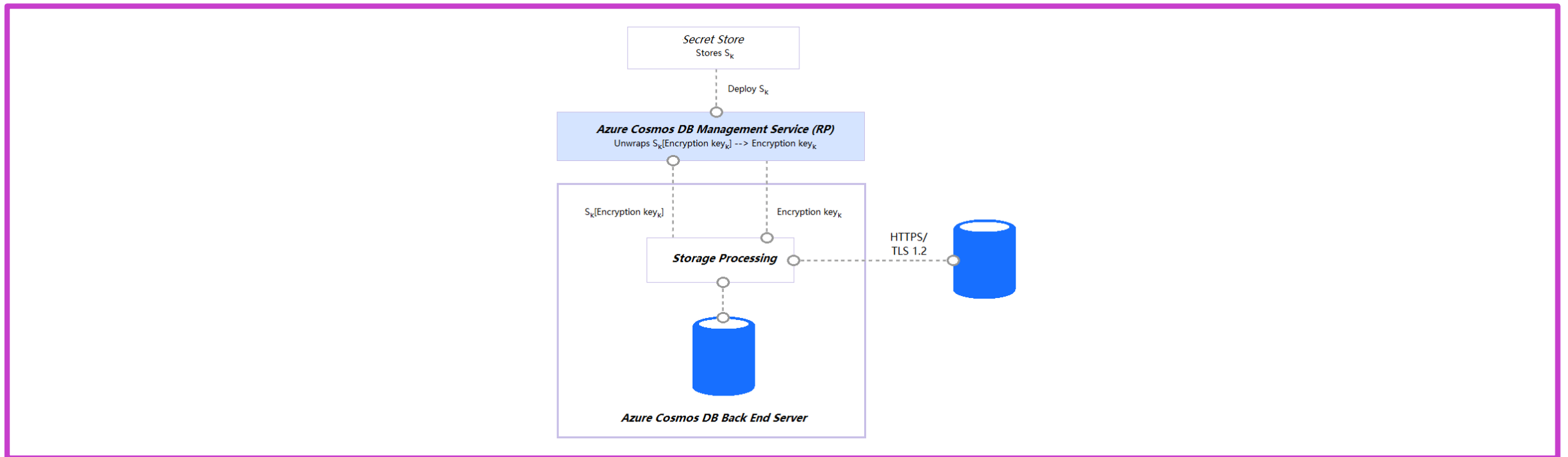
Troubleshoot issues with an IP access control policy

- Azure portal blocked
- SDK blocked
- Source IPs in blocked in requests
- Requests from a subnet with a service endpoint for Azure Cosmos DB enabled
- Private IP addresses in list of allowed addresses

Review data encryption options

Azure Cosmos DB now uses encryption at rest for all its databases, backups, and media. When Azure Cosmos DB data is in transit, or over the network, that data is also encrypted.

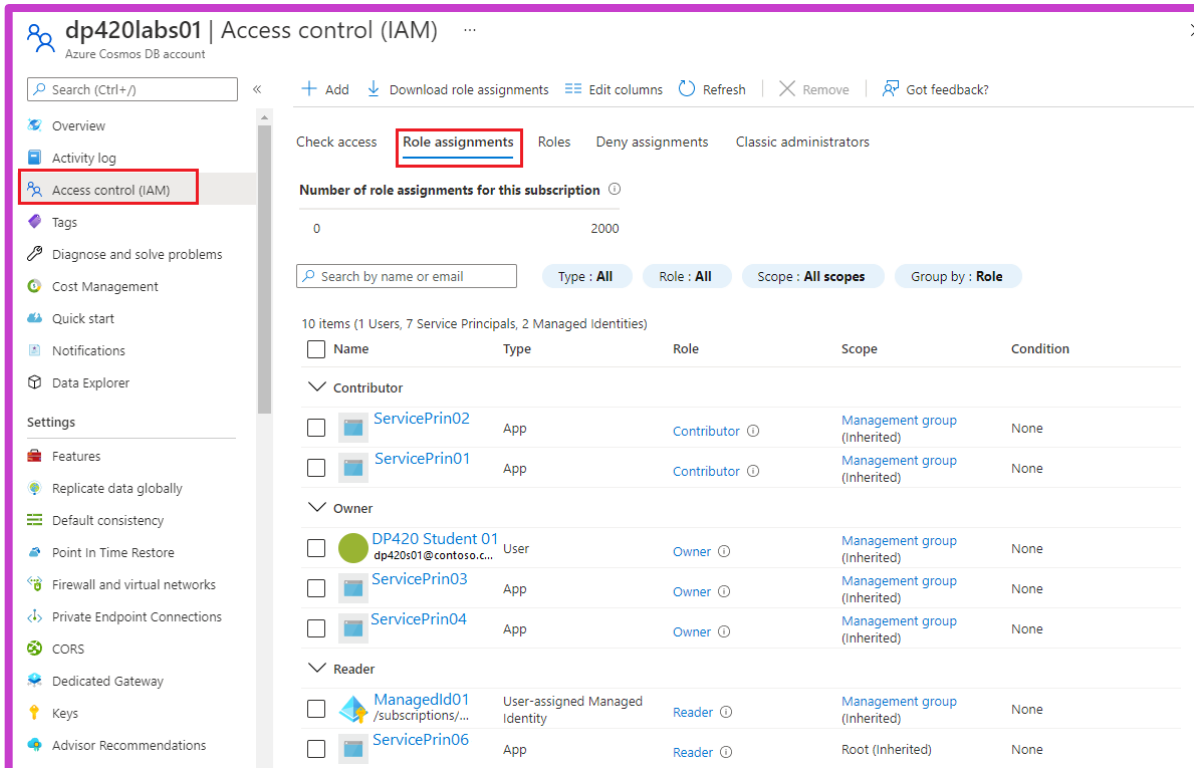
Azure Cosmos DB at rest and in transit encryption implementation



Use role-based access control (RBAC)

Azure role-based access control (RBAC) is provided in Azure Cosmos DB to do common management operations.

Identity and access management (IAM)



The screenshot displays the Azure portal's 'Access control (IAM)' page for a subscription named 'dp420labs01'. The 'Role assignments' tab is active, showing a list of 10 items (1 User, 7 Service Principals, 2 Managed Identities). The list includes roles like Contributor, Owner, and Reader assigned to various entities.

Name	Type	Role	Scope	Condition
Contributor				
ServicePrin02	App	Contributor	Management group (Inherited)	None
ServicePrin01	App	Contributor	Management group (Inherited)	None
Owner				
DP420 Student 01	User	Owner	Management group (Inherited)	None
ServicePrin03	App	Owner	Management group (Inherited)	None
ServicePrin04	App	Owner	Management group (Inherited)	None
Reader				
ManagedId01	User-assigned Managed Identity	Reader	Management group (Inherited)	None
ServicePrin06	App	Reader	Root (Inherited)	None

Other RBAC considerations

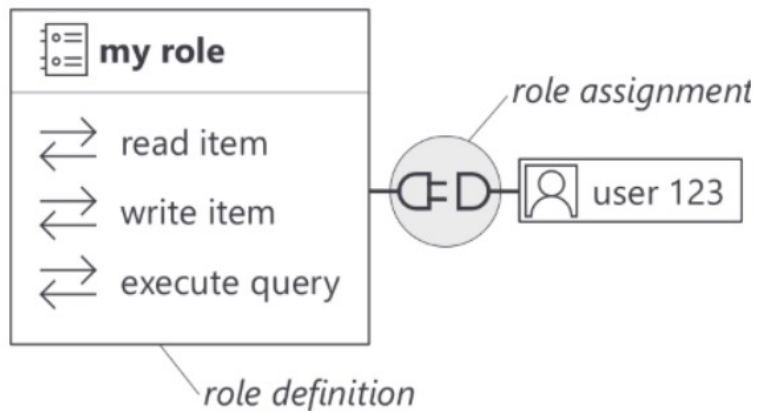
- Custom Controls
- Preventing changes from the Azure Cosmos DB SDKs

Access account resources using AAD

Access account resources using AAD allows you to authenticate your data requests with an Azure Active Directory (Azure AD) identity.

Permission model

```
Microsoft.DocumentDB/databaseAccounts/sqlDatabases/  
containers/*  
Microsoft.DocumentDB/databaseAccounts/sqlDatabases/  
containers/items/*  
Microsoft.DocumentDB/databaseAccounts/readMetadata
```



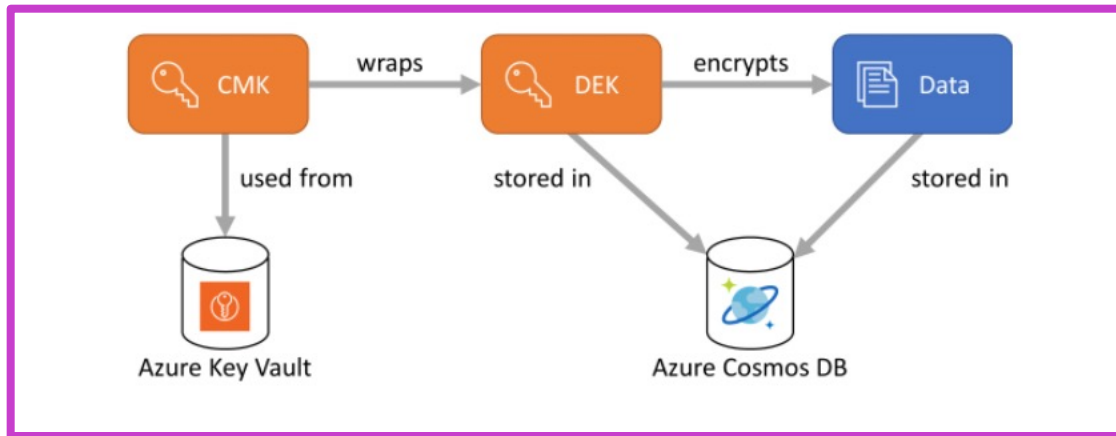
Initialize the SDK with Azure AD

```
TokenCredential servicePrincipal = new  
ClientSecretCredential(  
    "<azure-ad-tenant-id>",  
    "<client-application-id>",  
    "<client-application-secret>"  
);  
  
CosmosClient client = new CosmosClient("<account-  
endpoint>", servicePrincipal);
```

Understand Always Encrypted

Always encrypted encrypts sensitive data like credit card numbers or payroll information inside client-side applications.

Always encrypted concepts



Read encrypted items

```
var queryDefinition = container.CreateQueryDefinition(
    "SELECT * FROM c where c.property1 = @Property1"
);

await queryDefinition.AddParameterAsync(
    "@Property1",
    1234,
    "/property1"
);
```

Create a container with encryption policy

```
var path1 = new ClientEncryptionIncludedPath
{
    Path = "/property1",
    ClientEncryptionKeyId = "my-key",
    EncryptionType = EncryptionType.Deterministic.ToString(),
    EncryptionAlgorithm =
        DataEncryptionKeyAlgorithm.AEAD_AES_256_CBC_HMAC_SHA256.ToString()
};

var path2 = new ClientEncryptionIncludedPath
{
    Path = "/property2",
    ClientEncryptionKeyId = "my-key",
    EncryptionType = EncryptionType.Randomized.ToString(),
    EncryptionAlgorithm =
        DataEncryptionKeyAlgorithm.AEAD_AES_256_CBC_HMAC_SHA256.ToString()
};

await database.DefineContainer("my-container", "/partition-key")
    .WithClientEncryptionPolicy()
    .WithIncludedPath(path1)
    .WithIncludedPath(path2)
    .Attach()
    .CreateAsync();
```

Lab – Recover a database or container from a recovery point



- Prepare your development environment and create an Azure Cosmos DB for NoSQL account
- Create an Azure Key Vault and store the Azure Cosmos DB account credentials as a secret
- Create an Azure App Service webapp
- Import the multiple missing libraries into the .NET script
- Adding the Secret Identifier to your webapp
- (Optional) Install the Azure App Services Extension
- Deploy your application to Azure App Services
- Allow our app to use a managed identity
- Granting our web application an access policy to the Key Vault secrets

