Restore from Blob URL using Log Replay Service (LRS)

Last updated by | Radhika Shah | Jun 21, 2022 at 9:30 PM PDT

Contents

Issue

Customer restoring database on Azure SQL Managed Instance using Log Replay Service (LRS) appears stuck. Customer sees databases in 'Restoring' state.

Investigation/Analysis

Confirm that customer is restoring using LRS.

In ASC, check under Provisioning --> Operations --> Management Operations. LRS restores appears as **'CreateExternalManagedBackupRestoreV2Request'** under operation_Type column.

Below Kusto can be used to confirm the same (in the event that ASC is not available):

```
let serverName = '>MIName<';
            MonManagementOperations
            | where originalEventTimestamp >= {StartDateTime} and originalEventTimestamp <= {EndDateTime}
            | where event == 'management_operation_start'
            | where operation_parameters has serverName
            | extend InputParams = tostring(parse_xml(operation_parameters).InputParameters)
            | extend InputParams = iif(isempty(InputParams), operation_parameters, InputParams)
            | extend LogicalDatabaseName = parse_xml(operation_parameters).InputParameters.LogicalDatabaseName
            | extend DatabaseName = parse_xml(operation_parameters).InputParameters.DatabaseName
            | extend HubLogicalDatabaseName = parse_xml(operation_parameters).InputParameters.HubLogicalDataba
            | extend HubDatabaseName = parse_xml(operation_parameters).InputParameters.HubDatabaseName
            | extend SourceDatabase = parse_xml(operation_parameters).InputParameters.SourceDatabase
            | extend TargetDatabase = parse_xml(operation_parameters).InputParameters.TargetDatabase
            | extend Database = iif(isnotempty(database_name), database_name, '')
            | extend Database = iif(isempty(Database) and isnotempty(logical_database_name), logical_database_
            | extend Database = iif(isempty(Database) and isnotempty(LogicalDatabaseName), LogicalDatabaseName
            | extend Database = iif(isempty(Database) and isnotempty(DatabaseName), DatabaseName, Database)
            | extend Database = iif(isempty(Database) and isnotempty(HubLogicalDatabaseName), HubLogicalDataba
            | extend Database = iif(isempty(Database) and isnotempty(HubDatabaseName), HubDatabaseName, Databa
            | extend Database = iif(isempty(Database) and isnotempty(TargetDatabase), TargetDatabase, Database
            | extend Database = iif(isempty(Database) and isnotempty(SourceDatabase), SourceDatabase, Database
            | where operation_type != 'ExecuteDatabaseVulnerabilityAssessmentScan'
            | where operation_type != 'SendEmailNotificationsV2'
            | parse operation_parameters with * '<SubmitResourceHydration>' NotViaARM:bool '</SubmitResourceHy
            | extend NotRequestedViaARM = iif(isempty(NotViaARM),'', iif(NotViaARM == 1, 'True', 'False'))
            | project originalEventTimestamp, request_id, operation_type, operation_parameters, Database,Input
            | join kind = leftouter
            (
            MonManagementOperations
            | where originalEventTimestamp >= {StartDateTime}
            and originalEventTimestamp <= {EndDateTime} //remove to check for operations completing after EndT
            | where event != 'management_operation_start'
            | where operation_parameters has serverName
            | project originalEventTimestamp, request_id, event,error_code, error_message, operation_result
            ) on request_id
            | extend Result=replace(@'management_operation_*', @'\1', event)
            | extend OutputParams = tostring(parse_xml(operation_result).OutputParameters)
            | extend OutputParams = iif(isempty(OutputParams), operation_result, OutputParams)
            | project originalEventTimestamp, originalEventTimestamp1, Database, request_id, Result, operation
            | project-rename StartTime = originalEventTimestamp, EndTime = originalEventTimestamp1
```

On further investigating with the restore_id from above, additional error details can be identified:

```
MonRestoreEvents
| where restore_request_id contains "<restore_id guid>"
```

Sample Output from MonRestoreEvents kusto under details_with_customer_data column:

details_with_customer_data

Error happened while getting backup reference due to the following exception Microsoft.Xdb.Common. **Full backup can not be found.**\r\n at Microsoft.Xdb.BackupRestore.Common.BackupStore.AzureBackupRemoteStore.GetBackupReferences(StIExternalBackupBlobDirectoryBrowser externalBackupBlobDirectoryBrowser, List`1& unrestorableFilesSeBoolean isRetryOnRecoverableErrorsEnabled) in D:\dbs\sh\5uj5\0418_125232\cmd\d\sql\xdb\manifest\svc\BackupRestore\BRCommonV2\backupstore\Backup.cs:line 744.

Check the input_parameters for the operation (either in ASC Provisioning tab or via MonManagementOperations Kusto) to check the <BackupContainerUrl> parameter. The correct path to backup container url should be in the format:

```
https://<mystorageaccountname>.blob.core.windows.net/<containername>/<databasefolder>
```

Public document: https://docs.microsoft.com/en-us/azure/azure-sql/managed-instance/log-replay-service-migrate

If the <BackupContainerUrl> contains full path to .bak file (when using LRS via REST API), can lead to specified error.

Correct path:

```
https://<mystorageaccountname>.blob.core.windows.net/<foldername>
```

Incorrect path:

```
https://<mystorageaccountname>.blob.core.windows.net/<foldername>/<xyz.bak>
```

Note: When customer is restoring through SSMS, he is using Native Restore (via T-SQL) and not Log Replay. I see from telemetry that those restores completed, as said by customer. The Log replay restores failed due to the design of log replay:

Native Restore requires a direct path to .bak file. Hence the correct path would be:

```
https://<mystorageaccountname>.blob.core.windows.net/<foldername>/<xyz.bak>
```

But for log replay over rest API, customer should provide the path to the folder where the backup is (not the whole path to .bak file but the path to the folder where the .bak file is located) and the restore would work:

```
https://<mystorageaccountname>.blob.core.windows.net/<foldername>
```

Mitigation/Workaround

Have customer specify the correct path for backup in the format:

```
https://<mystorageaccountname>.blob.core.windows.net/<foldername>
```

Public Doc Reference

Log Replay Service

Internal Reference

ICM 306919320

How good have you found this content?