What is the Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases?

The Windows to Linux replatforming assistant for Microsoft SQL Server Databases service is a scripting tool that helps you move existing Microsoft SQL Server workloads from a Windows to a Linux operating system. You can use the replatforming assistant with any Windows Server virtual machines (VMs) hosted in the cloud or on-premises environments running SQL Server 2008 and above. The tool checks for common incompatibilities, exports database(s) from the Windows VM, and imports into an EC2 instance running SQL Server 2017 on Ubuntu 16.04. The automated process results in a ready-to-use Linux VM configured with your selected SQL Server database(s) that can be used for experimenting and testing.

Contents

What is the Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases? .............. 1

- Concepts ............................................................................................................................................ 2
- Backup .................................................................................................................................................. 2
- Restore ................................................................................................................................................ 2
- Replatform .......................................................................................................................................... 2
- Related Services ............................................................................................................................... 2

How Microsoft SQL Server Replatforming Works ............................................................................. 2

- Components ........................................................................................................................................ 3

Setting Up ............................................................................................................................................. 4

- Prerequisites ...................................................................................................................................... 4
  - Step 1: Install the AWS PowerShell Module ..................................................................................... 4
  - Step 2: Install the Windows to Linux Replatforming Assistant PowerShell Backup Script ..... 4
  - Step 3: Add an AWS User Profile to the AWS SDK Store ............................................................ 4
  - Step 4: Create an IAM Instance Profile Role .................................................................................. 5

Getting Started ...................................................................................................................................... 5

- Accessing the Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases .... 5

Running the Script ................................................................................................................................ 8

- Syntax ................................................................................................................................................ 8
- Example 1: Move a database to an EC2 instance ................................................................................. 8
- Example 2: Move a database to an EC2 instance using the AWS credential profile ....................... 8
Example 3: Move a database to a new m5.large type instance .................................................. 9
Example 4: Move all databases to a new m5.large type instance .................................................. 9
Release Notes .................................................................................................................................. 9
8 April 2019 ...................................................................................................................................... 9

Concepts

The following terminology and concepts are central to your understanding and use of the Windows to Linux replatforming assistant for Microsoft SQL Server Databases.

Backup

A Microsoft SQL Server backup copies data or log records from a Microsoft SQL Server database or its transaction log to a backup device, such as a disk. For more information, see Backup Overview (Microsoft SQL Server).

Restore

A logical and meaningful sequence for restoring a set of Microsoft SQL Server backups. For more information, see Restore and Recovery Overview (Microsoft SQL Server).

Replatform

A Microsoft SQL Server database can be replatformed from a Windows EC2 instance to a Linux EC2 instance running Microsoft SQL Server. It can also be replatformed to the VMware Cloud running Microsoft SQL Server Linux on AWS.

Related Services

AWS Systems Manager (SSM) gives you visibility and control of your infrastructure on AWS. The Windows to Linux replatforming assistant for Microsoft SQL Server Databases uses SSM to move your Microsoft SQL databases to Microsoft SQL Server on EC2 Linux. For more information on Systems Manager, see the AWS Systems Manager User Guide.

How Microsoft SQL Server Replatforming Works

Windows to Linux replatforming assistant for Microsoft SQL Server Databases allows you to migrate your Microsoft SQL Server databases from an on-premises environment or from an EC2 Windows instance to Microsoft SQL Server 2017 on EC2 Linux using backup/restore. For the destination EC2 Linux instance, you provide either the EC2 instance ID or the EC2 instance type with the subnet ID and EC2 Key Pair.
When you execute the PowerShell script for the Windows to Linux replatforming assistant for Microsoft SQL Server Databases on the source Microsoft SQL Server databases, the Windows instance backs up the databases to an encrypted Amazon Simple Storage Service (S3) storage bucket. It then launches a comparable Microsoft SQL on EC2 Linux instance, and imports the backups to a new instance. This process can be used to replatform your 2-tier databases running enterprise applications, and it enables you to replicate your database to Microsoft SQL on Linux to test the application while the source Microsoft SQL Server remains online. After testing, you can schedule application downtime and rerun the PowerShell backup script during your final cutover.

The entire replatforming process can also be automated and executed unattended. You can run the SSM automation document to import your existing database backup files into Microsoft SQL on EC2 Linux without using the PowerShell backup script.

**Components**

The Windows to Linux replatforming assistant for Microsoft SQL Server Databases script consists of two main components:

1. An SSM Automation document named AWSEC2-SQLServerDBRestore, which restores database backups to Microsoft SQL on EC2 Linux. This automation restores Microsoft SQL Server database backups stored in Amazon S3 to Microsoft SQL Server 2017 running on an EC2 Linux instance. You can provide your own EC2 instance running Microsoft SQL Server 2017 Linux, or the automation launches and configures a new EC2 instance with Microsoft SQL Server 2017 on Ubuntu 16.04. The automation supports the restoration of full, differential, and transactional log backups, and accepts multiple database backup files. The automation automatically restores the most recent valid backup of each database in the files provided. For more information, see AWSEC2-SQLServerDBRestore.

2. An AWS-signed PowerShell backup script, which can be used to back up on-premises Microsoft SQL Server databases to an Amazon S3 storage bucket.

You can integrate your existing backups with only the SSM Automation document to import the databases into Microsoft SQL on EC2 Linux. You can also use SSM Automation with the PowerShell backup script to back up the databases, transfer them to Amazon S3, and then automatically import the backups into Microsoft SQL on EC2 Linux.
Setting Up

Prerequisites

In order to run the Windows to Linux replatforming assistant for Microsoft SQL Server Databases script, you must:

- Install the AWS PowerShell module
- Install the PowerShell backup script
- Add one user profile to the AWS SDK store with permissions
- Create an AWS Identity and Access Management (IAM) instance profile role in order to run Systems Manager on your EC2 Linux instance

Step 1: Install the AWS PowerShell Module

To install the AWS PowerShell module, follow the steps listed at Setting up the AWS Tools for PowerShell on a Windows-Based Computer. We recommend that you use PowerShell 3.0 or later for the backup script to work properly.

Step 2: Install the Windows to Linux Replatforming Assistant PowerShell Backup Script

In order to run the Windows to Linux replatforming assistant, download the PowerShell backup script, MigrateSQLServerToEC2Linux.ps1.

Step 3: Add an AWS User Profile to the AWS SDK Store

To add and configure the AWS user profile, see the steps listed at Managing Profiles in the AWS Tools for PowerShell User Guide. Set the following IAM policy for your user profile. You can also add these permissions as an inline policy under your AWS user account using the IAM console.

```json
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:RebootInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeInstances",
      "ec2:CreateTags",
      "ec2:RunInstances",
      "ec2:DescribeImages",
      "iam:PassRole",
      "ssm:StartAutomationExecution",
      "ssm:DescribeInstanceInformation",
```
"ssm:ListCommands",
"ssm:ListCommandInvocations",
"ssm:SendCommand",
"ssm:GetAutomationExecution",
"s3:PutEncryptionConfiguration",
"s3:CreateBucket",
"s3:ListBucket",
"s3:PutObject",
"s3:DeleteObject"
],
   "Resource": "*
}]

Step 4: Create an IAM Instance Profile Role
To create an IAM instance profile role in order to run Systems Manager on EC2 Linux, see the steps listed under [Create an Instance Profile for Systems Manager](https://docs.aws.amazon.com/systems-manager/latest/userguide/) in the [AWS Systems Manager User Guide](https://docs.aws.amazon.com/systems-manager/latest/userguide/).

## Getting Started

### Accessing the Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases

The following parameters are used by the PowerShell script to replatform your Microsoft SQL Server databases.

**-SqlServerInstanceName**
The name of the Microsoft SQL Server instance to be backed up. If a value for `SqlServerInstanceName` is **not provided**, `$env:ComputerName` is **used by default**.
Type: String
Required: No

**-DBNames**
The names of the databases to be backed up and restored. Specify the names of the databases in a comma-separated list (for example, `adventureDB, universityDB`). Either the `DBNames` or `MigrateAllDB` parameter is required.
Type: Object []
Required: No
-MigrateAllDBs
This switch is disabled by default. If this switch is enabled, the automation migrates all databases except for the system databases (master, msdb, tempdb). Either the DBNames or MigrateAllDB parameter is required.
Type: SwitchParameter
Required: No

-PathForBackup
The path where the full backup is stored.
Type: String
Required: Yes

-SetSourceDBModeReadOnly
This switch is disabled by default. If this switch is enabled, it makes the database read-only during migration.
Type: SwitchParameter
Required: No

-IamInstanceProfileName
Enter the AWS IAM instance role with permissions to run SSM Automation on your behalf. See Configuring Access for Systems Manager Automation.
Type: String
Required: Yes

-AWSRegion
Enter the AWS region where your Amazon S3 buckets are created to store database backups.
Type: String
Required: Yes

-EC2InstanceId
To restore Microsoft SQL Server databases to an existing EC2 instance running Microsoft SQL Server Linux, enter the instance ID of the instance. Make sure that the EC2 instance already has the AWS SSM agent installed and running.
Type: String
Required: No
- **EC2InstanceType**
  To restore Microsoft SQL Server databases to a new EC2 Linux instance, enter the instance type of the instance to be launched.
  Type: String
  Required: No

- **EC2KeyPair**
  To restore Microsoft SQL Server databases to a new EC2 Linux instance, enter the name of the EC2 Key Pair to be used to access the instance. This parameter is recommended if you are creating a new EC2 Linux instance.
  Type: String
  Required: No

- **SubnetId**
  This parameter is required when creating a new EC2 Linux instance. When creating a new EC2 Linux instance, if SubnetId is not provided, the AWS user default subnet is used to launch the EC2 Linux instance.
  Type: String
  Required: No

- **AWSProfileName**
  The name of the AWS profile the automation uses when connecting to AWS services. See [Configuring Access for Systems Manager Automation](#) for more information on the required IAM user permissions. If a profile is not entered, the automation uses your default AWS profile.
  Type: String
  Required: No

- **AWSProfileLocation**
  The location of the AWS Profile if the AWS Profile is not stored in the default location.
  Type: String
  Required: No

- **GeneratePresignedUrls**
  This parameter is only used when replatforming to non-EC2 instances, such as to VMware Cloud on AWS or on-premises VMs.
  Type: SwitchParameter
  Required: No
<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see About Common Parameters in the Microsoft PowerShell documentation.

Required: No

Running the Script

The following common scenarios and example PowerShell scripts demonstrate how to replatform your Microsoft SQL Server databases using Windows to Linux re platforming assistant for Microsoft SQL Server Databases.

Syntax

The Windows to Linux re platforming assistant for Microsoft SQL Server Databases script adheres to the following syntax:

```
```

Example 1: Move a database to an EC2 instance

This example shows how to move a database named AdventureDB to an EC2 Microsoft SQL Server on Linux instance, with an instance ID of -i-024689abcdef, from the Microsoft SQL Server Instance named MSSQLSERVER. The backup directory to be used is D:\Backup and the AWS region is us-east-2.

```
PS C:\> ./MigrateSqlServerToEC2Linux.ps1 -SqlServerInstanceName MSSQLSERVER -EC2InstanceId i-024689abcdef -DBNames AdventureDB -PathForBackup D:\Backup -AWSRegion us-east-2 -IamInstanceProfileName AmazonEC2RoleForSSM
```

Example 2: Move a database to an EC2 instance using the AWS credential profile

This example shows how to move the database in Example 1 using the AWS cred profile: DBMigration.

```
PS C:\> ./MigrateSqlServerToEC2Linux.ps1 -SqlServerInstanceName MSSQLSERVER -EC2InstanceId i-024689abcdef -DBNames AdventureDB -PathForBackup D:\Backup -AWSRegion us-east-2 -AWSProfileName DBMigration -IamInstanceProfileName AmazonEC2RoleForSSM
```
Example 3: Move a database to a new m5.large type instance
This example shows how to create an m5.large type EC2 Linux instance in subnet-abc127 using the Key Pair customer-ec2-keypair and then moving AdventureDB and TestDB to the new instance from the database used in Examples 1 and 2.

```bash
PS C:\> ./MigrateSQLServerToEC2Linux.ps1 -EC2InstanceType m5.large -SubnetId subnet-abc127 -EC2KeyPair customer-ec2-keypair -DBNames AdventureDB,TestDB -PathForBackup D:\Backup -AWSRegion us-east-2 -AWSProfileName DBMigration -IamInstanceProfileName AmazonEC2RoleForSSM
```

Example 4: Move all databases to a new m5.large type instance
This example shows how to create an m5.large type EC2 Linux instance in subnet-abc127 using the Key Pair customer-ec2-keypair and then migrating all databases to the instance from databases used in Examples 1 and 2.

```bash
PS C:\> ./MigrateSQLServerToEC2Linux.ps1 -EC2InstanceType m5.large -SubnetId subnet-abc127 -EC2KeyPair customer-ec2-keypair -MigrateAllDBs -PathForBackup D:\Backup -AWSRegion us-east-2 -AWSProfileName DBMigration -IamInstanceProfileName AmazonEC2RoleForSSM
```

**Release Notes**

8 April 2019

Currently, Windows to Linux replatforming assistant for Microsoft SQL Server Databases only supports Microsoft SQL Server 2017 for restoring databases on Microsoft SQL Server on EC2 Linux.