|  |  |
| --- | --- |
| **Key Note** | **Description** |
| Process Name | TM VSS Backup & Refresh Guide |
| Process Owner | Naveen Gollamudi |
| Process Description | TMG Guide to brief VSS Backup & Restore/Refresh process |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Change Date** | **Changes** | **Changer** | **Version** |
| 03/11/2020 | Initial version added | Naveen Gollamudi | 1.0 |

Contents

[Overview 3](#_Toc55305248)

[VSS Pre-Requirements 3](#_Toc55305249)

[1. EC2 Endpoint access 3](#_Toc55305250)

[2. IAM Role for AWS Backup 3](#_Toc55305251)

[3. IAM Permissions for EC2 IAM Role 4](#_Toc55305252)

[4. AWS CLI 5](#_Toc55305253)

[VSS Backup 5](#_Toc55305254)

[1. KMS Key for Backup vault 5](#_Toc55305255)

[2. Backup Vault 5](#_Toc55305256)

[3. Backup Plan 5](#_Toc55305257)

[4. AWS Backup Assign Resource 6](#_Toc55305258)

[5. Adhoc VSS Snapshots 6](#_Toc55305259)

[Database Objects 7](#_Toc55305260)

[1. Schema 7](#_Toc55305261)

[2. Table(s) 7](#_Toc55305262)

[3. Stored Procedures 7](#_Toc55305263)

[Refresh Configuration 7](#_Toc55305264)

[1. VSS.AttachUserDB\_RefreshConfiguration 7](#_Toc55305265)

[2. VSS.AttachUserDB\_RefreshConfigurationFiles 8](#_Toc55305266)

[Update Master Files 8](#_Toc55305267)

[VSS Refresh 8](#_Toc55305268)

[Validate Refresh 9](#_Toc55305269)

[Pricing 9](#_Toc55305270)

# Overview

This document is to cover the VSS Snapshots for backing up SQL Server(s) including all the volumes (including root), Refreshing set of databases from one (prod) environment to other environments (non-prod) to provide the live copy of the data for development, test and support environments.

# VSS Pre-Requirements

In order to configure VSS Backup/Restore, below are the pre-requisites.

## EC2 Endpoint access

To allow access to EC2 endpoint “com.amazonaws.eu-west-1.ec2” over port 443 (https), follow below steps.

**EC2 Endpoint:** Create endpoint under VPC Console, for the region EC2 are hosted.

Ref: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/interface-vpc-endpoints.html>

**Security Group & Rule:** Once endpoint is created, ensure the EC2 instances has a egress rule to the Security group associated with this endpoint on port 443.

## IAM Role for AWS Backup

Create IAM role to configure AWS Backups, make sure the Role name contains “AWSBackup” \* to start with, and attach “[AWSBackupServiceRolePolicyForBackup](https://console.aws.amazon.com/iam/home?region=eu-west-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2Fservice-role%2FAWSBackupServiceRolePolicyForBackup)” policy (This is AWS Policy), either through console or using below terraform code.

#IAM Role for AWS Backup

resource "aws\_iam\_role" "vssrole" {

  name               = "AWSBackup\_vss\_role"

  assume\_role\_policy = data.aws\_iam\_policy\_document.vsspolicy.json

}

data "aws\_iam\_policy\_document" "vsspolicy" {

  statement {

    actions = ["sts:AssumeRole"]

    principals {

      type        = "Service"

      identifiers = ["backup.amazonaws.com"]

    }

  }

}

resource "aws\_iam\_role\_policy\_attachment" "vsspolatcmnt" {

  role       = aws\_iam\_role.vssrole.name

  policy\_arn = "arn:aws:iam::aws:policy/service-role/AWSBackupServiceRolePolicyForBackup"

}

\*(As this role would be used with other role which has “iampassrole” permissions for the roles start with “AWSBackup” for Adhoc backups and restores).

## IAM Permissions for EC2 IAM Role

Create the policy as scripted below and attach it to the IAM role for all EC2 instances involved in Backup & Restore. Need to attach one more policy “AWSBackupOperatorAccess” (This is AWS Policy) to all the EC2 instances involved.

#Create IAM policy for VSS Snapshots

resource "aws\_iam\_policy" "ec2vsspolicy" {

  name   = "ec2-vsspolicy"

  policy = <<-EOF

  {

    "Version": "2012-10-17",

    "Statement": [

        {

            "Effect": "Allow",

            "Action": "ec2:CreateTags",

            "Resource": "arn:aws:ec2:\*::snapshot/\*"

        },

        {

            "Effect": "Allow",

            "Action": [

                "ec2:DescribeInstances",

                "ec2:CreateSnapshot",

                "ec2:CreateTags",

                "ec2:CreateVolume",

                "ec2:AttachVolume",

                "ec2:DescribeVolume\*",

                "ec2:Describe\*",

                "ec2:DescribeSnapshots",

                "ec2:DetachVolume",

                "ec2:DeleteVolume",

                "kms:CreateGrant",

                "kms:ListGrants",

                "kms:RevokeGrant",

                "kms:Encrypt",

                "kms:Decrypt",

                "kms:ReEncrypt\*",

                "kms:GenerateDataKey\*",

                "kms:DescribeKey"

            ],

            "Resource": "\*"

        }

    ]

    }

  EOF

}

## AWS CLI

Install latest AWS CLI on all EC2 involved, min version required : 2.0.56

# VSS Backup

## KMS Key for Backup vault

Create KMS key to use for Backup vault, where all backups are stored either using AWS console or below terraform code.

#creating kms customer key for VSS backup vault

resource "aws\_kms\_key" "vss\_backup" {

  description             = "KMS key for  vss snapshots"

  deletion\_window\_in\_days = 7 #days key will be deleted after destruction of the resource

  tags = {

    Name        = "vss\_backup"

    Description = "KMS key for VSS snapshots"

  }

}

resource "aws\_kms\_alias" "vss\_backup\_alais" {

  name          = "alias/vss\_backup"

  target\_key\_id = aws\_kms\_key.vss\_backup.key\_id

}

## Backup Vault

Create the Backup Vault with the KMS key created in previous step, either using console or below terraform code. Define local tags, if you need more tags otherwise remove “tags = local.tags”

#Backup Vault

resource "aws\_backup\_vault" "vssbackup" {

  name        = "dwhtest\_vss\_vault"

  kms\_key\_arn = aws\_kms\_key.vss\_backup.arn

tags = local.tags

}

## Backup Plan

Create the Backup Plan and configure to use the Vault created earlier. Define local tags, if you need more tags otherwise remove “tags = local.tags” and “recovery\_point\_tags = local.tags”

resource "aws\_backup\_plan" "vss\_plan" {

  name = "vss\_backupplan"

  advanced\_backup\_setting {

    backup\_options = {

      "WindowsVSS" = "enabled"

    }

    resource\_type = "EC2"

  }

  rule {

    rule\_name           = "DailyBackups"

    target\_vault\_name   = aws\_backup\_vault.vssbackup.name

    schedule            = "cron(0 21 ? \* \* \*)"

    recovery\_point\_tags = local.tags

    lifecycle {

      delete\_after = 7

    }

  }

tags = local.tags

}

In this plan, could customise the Cron Schedule to change the scheduler times and life cycle to define backup retention policy accordingly.

## AWS Backup Assign Resource

Once all the above steps are completed, need to assign the EC2 resource to take VSS Snaphots.

resource "aws\_backup\_selection" "vssbackupassign" {

  name         = "<prodec2server>"

  iam\_role\_arn = aws\_iam\_role.vssrole.arn

  plan\_id      = aws\_backup\_plan.vss\_plan.id

  resources    = [aws\_instance.<PRODEC2Server>.arn]

}

In the above script, need to replace with the IAM Role created for AWS Backup, Backup plan and EC2 instance that needs backing up using VSS Snapshots

## Adhoc VSS Snapshots

Above configuration ensures backups are running on schedule, however when we need an Adhoc backup need to run the below PowerShell Script from the Source server, this can be created as a job, so it can be called from Octopus/Devops life cycle (EX: Prior to major release)…need to fill the appropriate config details.

aws backup start-backup-job --backup-vault-name <VaultName> --resource-arn arn:aws:ec2:<Region>:<AWSAccountNumber>:instance/<InstanceID> --backup-options WindowsVSS=enabled --iam-role-arn <IAMRoleCreatedforVSSBackupARN> --start-window-minutes 60 --complete-window-minutes 10080 --lifecycle DeleteAfterDays=30 --region <Region>

# Database Objects

In order to setup Restore\Refresh process, need to create below database objects and configure them accordingly.

## Schema

Choose the database to be used to create all the objects and create a new schema “VSS” , Can use below script to create schema



## Table(s)

Need to create few table(s) in order configure the Restore\Refresh and other required data.



## Stored Procedures

Create below Stored Procedures using the script.



# Refresh Configuration

Once all Database objects are created, need to configure Refresh information in below Two tables

## VSS.AttachUserDB\_RefreshConfiguration

Need to create a new row in this table to define all the required parameters of the Refresh, for Target and Source Servers, if both Data and log values are same please provide same for both the fields.

**RefreshName** – Unique name to define the refresh

**DataPath** – This is the path for the data disk to be used (AWS EX: xvdg, last character is disk letter)

**DataDisk** – Disk Letter to be used for data disk

**LogPath** – This is the path for the Log disk to be used (AWS EX: xvdh, last character is disk letter)

**LogDisk** – Disk Letter to be used for Log disk

**EBSKMS** – ARN of the KMS key to be used to encrypt newly provided volumes

**UseRecent –** (bit) To use recent snapshot use 1 (this overrides below snapshot id(s), if it is set to 0 need to provide below snapshot id(s)

**SourceDataSnapID –** If theUseRecent set to 0, need to provide specific Data snapshot ID to be restored

**SourceLogSnapID** - If theUseRecent set to 0, need to provide specific Log snapshot ID to be restored

**SourceDataVol** – Source Data Volume ID, to be used to restore to target server

**SourceLogVol** – Source Log Volume ID, to be used to restore to target server

**SourceServer** – Source SQL Server instance to capture database file information

## VSS.AttachUserDB\_RefreshConfigurationFiles

This table is to provide the list of databases to be restored as part of the Refresh configured in above step, add each database in separate row.

**DatabaseName** -- Name of the database(s) to be restored

**Prefix (Optional) –** If database(s) to be attached with a prefix on target server

**Suffix (Optional) -** If database(s) to be attached with a suffix on target server

# Update Master Files

Once Above two tables are configured, the PowerShell Script(s) below to be run to update the database file(s) information source server onto Target Server, as these file information is required in order to attach the database(s).

“Update-MasterFiles.ps1” imports and invokes “Update-MasterFiles.psm1”, Ensure the path of the .psm1 file is reflecting in .ps1 file.



This PS Script updates the database(s) File information in “VSS.AttachUserDB\_MasterFiles” table.

# VSS Refresh

This is the key part of the Refresh process, below PS Script needs to be invoked with the refresh name as input parameter to initiate the Refresh process.



Recommended to use SQL Job Step, to run as operating system (CmdExec) to run with PS latest version else it would run as PS V 1.0.

Ensure the PS script is stored on target server and path is correct.

Example Invocation : PowerShell –Command "& { S:\Powershell\VSSRestore.ps1 -RefreshName 'Warehouse' }"

# Validate Refresh

Once the Refresh is completed, Below SP can be used (this was created as part object(s) creation) to validate Refresh ran fine and all database(s) required are attached with the correct RefreshName.

EXEC VSS.uspCheckRefreshSuccessful @RefreshName = 'Warehouse'

# Pricing

VSS Snapshots captures only block level changes, so the pricing for snapshots is for the change rates not for the whole volume for every time Snapshot ran.

**Snapshot costs** : $ 0.05 /GB/Month in Ireland location when this document is written.

**Pricing Example:**

Total Size of All Volume(s) on Source Server **: 1024 GB**

Snapshot Retention **: 14 Snapshots/Days**

Change Rate **: 10% /Day**

Restored Volume(s) Size : **400 GB**

**Snapshot Pricing : 1024 \* 0.05 – for first snapshot = $51**

**100 \* 0.05 \* 13 -for next snapshots 10% change rate = $65**

**Target Volume Pricing (GP2 in Ireland) : 400 \* 0.11 = $44**