Cross Region Copy of VM Restore Points

Contents

[Overview 2](#_Toc68766612)

[Glossary 2](#_Toc68766613)

[High-level sequence 2](#_Toc68766614)

[Cross-region copy of VM Restore Points 3](#_Toc68766615)

[Create Restore Point Collection in target region 3](#_Toc68766616)

[URI Request 3](#_Toc68766617)

[Request Body 3](#_Toc68766618)

[Request Response 3](#_Toc68766619)

[Create VM Restore Point in Target Region 4](#_Toc68766620)

[URI Request 4](#_Toc68766621)

[Request Body 4](#_Toc68766622)

[Request Response 4](#_Toc68766623)

[Get VM Restore Points Copy/Replication Status 7](#_Toc68766624)

[URI Request 7](#_Toc68766625)

[Request Response 7](#_Toc68766626)

Overview

As an extension to [VM Restore Points](https://github.com/Azure/Virtual-Machine-Restore-Points) we are providing additional functionality within Azure platform to enable our partners to build BCDR solutions for Azure VMs. One such functionality is:

**Ability to copy VM Restore Points from one region to another other region**

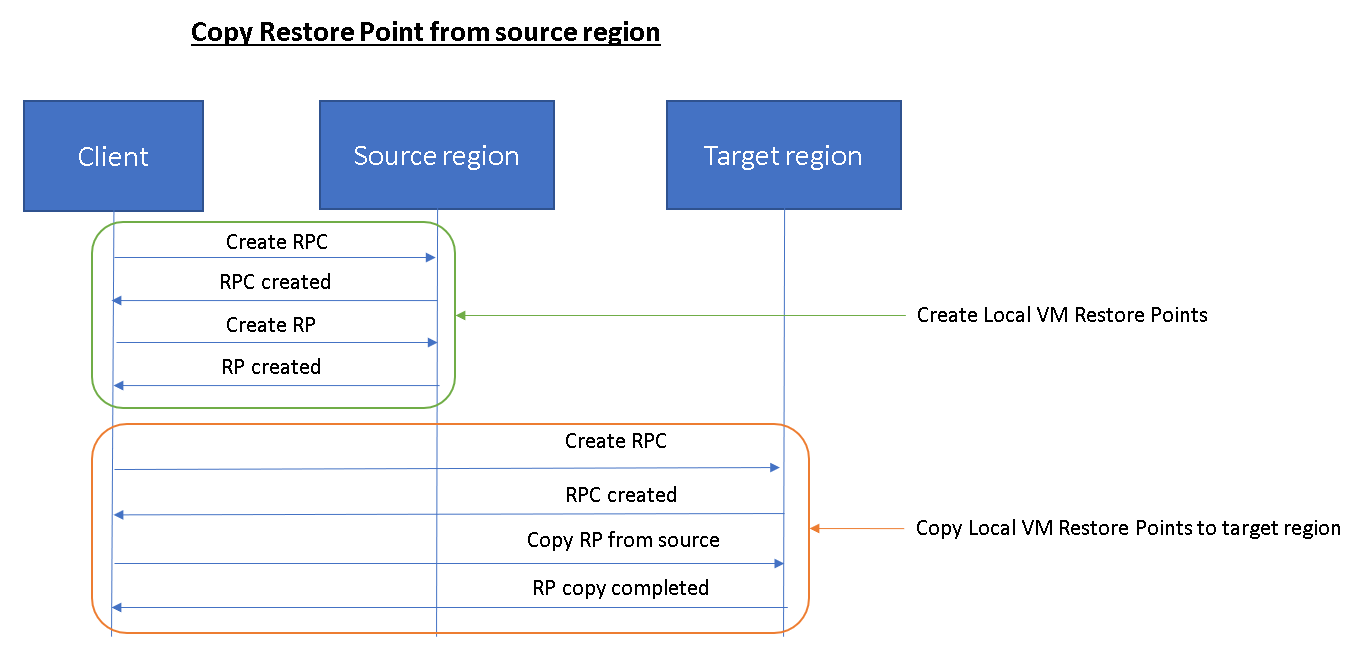
* Scenarios where this API can be helpful:
  + Extend multiple copies of backup to different regions
  + Extend local backup solutions to support disaster recovery from region failures
* Tentative Private Preview Availability: May 2021

This document will provide you with necessary information about the API surface for copying local restore points to a target region so that you can get started to build your BCDR solution for Azure VMs.

Glossary

* Ubiquitous DR  Platform support for enabling DR solution to be available for every Azure IaaS VM with managed disks
* VM Restore Points (RP)  VM level multi-disk/volume consistent snapshots
* Restore Point Collection (RPC)  A logical container for all the Restore Points of a particular VM.
* Remote Restore Points  Restore Points in a different region as compared to the region of the source VM

High-level sequence

**NOTE**: For copying a RestorePoint across region, you need to pre-create a RestorePoint in the local region. For details regarding creation of VM Restore Points in the local region refer to the [VM Restore Points documentation](https://github.com/Azure/Virtual-Machine-Restore-Points).

Cross-region copy of VM Restore Points

Create Restore Point Collection in target region

First step in copying an existing VM Restore point from one region to another is to create a RestorePointCollection in the target region by referencing the RestorePointCollection from the source region.

URI Request

PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/restorePointCollections/{restorePointCollectionName}&api-version={api-version}

Request Body

{

"name": "name of the copy of restorePointCollection resource",

"location": "location of the copy of the restorePointCollection resource",

"tags": {

"department": "finance"

},

"properties": {

"source": {

"id": "/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microsoft.compute/restorePointCollections/{restorePointCollectionName}"

}

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Mandatory | Type | Description | |
| name | Yes | String | Name of the restore point collection in target region |
| location | Yes | String | Location of the restore point collection (Target region) |
| tags | No | Object | Tags for the restore point collection |
| properties.source.id | Yes | String | ARM id of the original RestorePointCollection in source region |

Request Response

The request response will include a status code and set of response headers.

*Status Code*

The operation returns a 201 during create and 200 during Update.

*Response body*

{

"name": "name of the copied restorePointCollection resource",

“id”: “CSM Id of copied restorePointCollection resource”,

"type": "Microsoft.Compute/restorePointCollections",

"location": "location of the copied restorePointCollection resource",

"tags": {

"department": "finance"

},

"properties": {

"source": {

"id": "/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microsoft.compute/restorePointCollections/{restorePointCollectionName}"

"location" : “location of source RPC”

}

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Type | Description | |
| name | String | Name of the restore point collection in target region |
| id | String | Location of the restore point collection (Target region) |
| type | String | Resource type: Microsoft.Compute/restorePointCollections |
| location | String | Location of the restore point collection (Target region) |
| tags | Object | Tags for the restore point collection |
| properties.source.location | String | Location of the source RPC. This is a read-only property returned as part of PUT / GET responses |

Create VM Restore Point in Target Region

Next step is to trigger creation of a RestorePoint in the target RestorePointCollection referencing the RestorePoint in the source region that needs to be copied.

URI Request

PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/restorePointCollections/{restorePointCollectionName}/restorePoints/{restorePointName}&api-version={api-version}

Request Body

{

"name": "name of the restore point resource",

"sourceRestorePoint": {

"id": "/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microsoft.compute/restorePointCollections/{restorePointCollectionName}/restorePoints/{restorePointName}"

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Mandatory | Type | Description | |
| name | Yes | String | Name of the restore point collection in target region |
| sourceRestorePoint.id | Yes | String | ARM id of the original RestorePoint in source region |

**NOTE:** Location of the sourceRestorePoint would be inferred from that of the source RestorePointCollection

### Request Response

The request response will include a status code and set of response headers.

#### Status Code

This is a long running operation; hence the operation returns a 201 during create. The client is expected to poll for the status using the operation. (Both the Location and Azure-AsyncOperation headers are provided for this purpose.)

During restore point creation, the ProvisioningState would appear as Creating in GET restore point API response. If creation fails, its ProvisioningState will be Failed. ProvisioningState would be set to Succeeded when the data copy across regions is initiated.

**NOTE**: The status of data movement can be tracked via “InstanceView” property. Restore Point is considered usable (can be used to restore a VM) only when copy of all the disk restore points are successful.

*Response body*

{

"id": "CSM Id of the restore point",

"name": "name of the restore point",

"optionalProperties": "opaque bag of properties to be passed to extension"

"sourceRestorePoint": {

"id": "/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microsoft.compute/restorePointCollections/{restorePointCollectionName}/restorePoints/{restorePointName}"

},

"consistencyMode": "CrashConsistent | FileSystemConsistent | ApplicationConsistent",

"sourceMetadata": {

"vmId": "Unique Guid of the VM from which the restore point was created",

"location" : "source VM location"

"hardwareProfile": {

"vmSize": "Standard\_A1"

},

"osProfile": {

"computername": "",

"adminUsername": "",

"secrets": [

{

"sourceVault": {

"id": "/subscriptions/<subId>/resourceGroups/<rgName>/providers/Microsoft.KeyVault/vaults/<keyvault-name>"

},

"vaultCertificates": [

{

"certificateUrl": "https://<keyvault-name>.vault.azure.net/secrets/<secret-name>/<secret-version>",

"certificateStore": "certificateStoreName on Windows"

}

]

}

],

"customData": "",

"windowsConfiguration": {

"provisionVMAgent": "true|false",

"winRM": {

"listeners": [

{

"protocol": "http"

},

{

"protocol": "https",

"certificateUrl": ""

}

]

},

"additionalUnattendContent": [

{

"pass": "oobesystem",

"component": "Microsoft-Windows-Shell-Setup",

"settingName": "FirstLogonCommands|AutoLogon",

"content": "<XML unattend content>"

}

],

"enableAutomaticUpdates": "true|false"

},

"linuxConfiguration": {

"disablePasswordAuthentication": "true|false",

"ssh": {

"publicKeys": [

{

"path": "Path-Where-To-Place-Public-Key-On-VM",

"keyData": "PEM-Encoded-public-key-file"

}

]

}

}

},

"storageProfile": {

"osDisk": {

"osType": "Windows|Linux",

"name": "OSDiskName",

"diskSizeGB": "10",

"caching": "ReadWrite",

"managedDisk": {

"id": "CSM Id of the managed disk",

"storageAccountType": "Standard\_LRS"

},

"diskRestorePoint": {

"id": "/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpcName>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>",

}

},

"dataDisks": [

{

"lun": "0",

"name": "datadisk0",

"diskSizeGB": "10",

"caching": "ReadWrite",

"managedDisk": {

"id": "CSM Id of the managed disk",

"storageAccountType": "Standard\_LRS"

},

"diskRestorePoint": {

"id": "/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpcName>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>",

}

}

]

},

"diagnosticsProfile": {

"bootDiagnostics": {

"enabled": true,

"storageUri": " http://storageaccount.blob.core.windows.net/"

}

},

},

"provisioningState": "Succeeded | Failed | Creating | Deleting",

"provisioningDetails": {

"creationTime": "Creation Time of Restore point in UTC",

}

}

|  |  |  |
| --- | --- | --- |
| Element Name | Type | Description |
| Name | String | Specifies the name of the restore point. |
| consistencyMode | String | Currently this is just an output property specified by the server. In future, client can also specify this value to indicate the type of restore point to be created. |
| sourceMetadata | Object | Properties of the resource from/for which the restore point was created. These properties are captured at the time of creation of the restore point |
| provisioningState | String | ProvisioningState of the copied RestorePoint resource. We would mark the ProvisioningState as Terminal, once we kick-off DataMove for that restore point, since data move is a long-running process. The status of data movement can be tracked via “InstanceView” properties. |
| provisioningDetails | Object | A set of server generated properties. These are specific to local RestorePoints. These values are not set in case of copy of the RestorePoint. |
| disk.diskRestorePoint.id | String | A unique id for each Disk attached to the VM |
| instanceView | Object | Contains the Aggregate Replication State of the RestorePoint from source to the target region as well as replication status of individual Disk Restore Points |

## Get VM Restore Points Copy/Replication Status

Once copy of VM Restore Points is initiated, you can track the copy status by calling GET instance View (?$expand=instanceView) on the target VM Restore Point.

### URI Request

GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/restorePointCollections/{restorePointCollectionName}/restorePoints/{restorePointName}?$expand=instanceView&api-version={api-version}

### Request Response

{

"id": "CSM Id of the restore point",

"name": "name of the restore point",

"optionalProperties": "opaque bag of properties to be passed to extension"

"sourceRestorePoint": {

"id": "/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microsoft.compute/restorePointCollections/{restorePointCollectionName}/restorePoints/{restorePointName}"

},

"consistencyMode": "CrashConsistent | FileSystemConsistent | ApplicationConsistent",

"sourceMetadata": {

"vmId": "Unique Guid of the VM from which the restore point was created",

"location" : "source VM location"

"hardwareProfile": {

"vmSize": "Standard\_A1"

},

"osProfile": {

"computername": "",

"adminUsername": "",

"secrets": [

{

"sourceVault": {

"id": "/subscriptions/<subId>/resourceGroups/<rgName>/providers/Microsoft.KeyVault/vaults/<keyvault-name>"

},

"vaultCertificates": [

{

"certificateUrl": "https://<keyvault-name>.vault.azure.net/secrets/<secret-name>/<secret-version>",

"certificateStore": "certificateStoreName on Windows"

}

]

}

],

"customData": "",

"windowsConfiguration": {

"provisionVMAgent": "true|false",

"winRM": {

"listeners": [

{

"protocol": "http"

},

{

"protocol": "https",

"certificateUrl": ""

}

]

},

"additionalUnattendContent": [

{

"pass": "oobesystem",

"component": "Microsoft-Windows-Shell-Setup",

"settingName": "FirstLogonCommands|AutoLogon",

"content": "<XML unattend content>"

}

],

"enableAutomaticUpdates": "true|false"

},

"linuxConfiguration": {

"disablePasswordAuthentication": "true|false",

"ssh": {

"publicKeys": [

{

"path": "Path-Where-To-Place-Public-Key-On-VM",

"keyData": "PEM-Encoded-public-key-file"

}

]

}

}

},

"storageProfile": {

"osDisk": {

"osType": "Windows|Linux",

"name": "OSDiskName",

"diskSizeGB": "10",

"caching": "ReadWrite",

"managedDisk": {

"id": "CSM Id of the managed disk",

"storageAccountType": "Standard\_LRS"

},

"diskRestorePoint": {

"id": "/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpcName>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>",

}

},

"dataDisks": [

{

"lun": "0",

"name": "datadisk0",

"diskSizeGB": "10",

"caching": "ReadWrite",

"managedDisk": {

"id": "CSM Id of the managed disk",

"storageAccountType": "Standard\_LRS"

},

"diskRestorePoint": {

"id": "/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpcName>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>",

}

}

]

},

"diagnosticsProfile": {

"bootDiagnostics": {

"enabled": true,

"storageUri": " http://storageaccount.blob.core.windows.net/"

}

},

},

"provisioningState": "Succeeded | Failed | Creating | Deleting",

"provisioningDetails": {

"creationTime": "Creation Time of Restore point in UTC",

},

"instanceView": {

"statuses": [{

"code": "ReplicationState/succeeded",

"level": "Info",

"displayStatus": "Replication succeeded",

"message": "<will be populated in error scenarios>",

"time": "2019-10-14T21:29:47.477089+00:00"

}],

"diskRestorePoints": [{

"id": "<diskRestorePoint Arm Id>",

“replicationStatus” : {

"status": {

"code": "ReplicationState/succeeded",

"level": "Info",

"displayStatus": "Replication succeeded",

"message": "<will be populated in error scenarios>",

"time": "2019-10-14T21:29:47.477089+00:00"

}

“completionPercent” : <completion percentage of the replication>

}

}],

}

}

|  |  |  |
| --- | --- | --- |
| Element Name | Type | Description |
| instanceView | Object | Contains the Aggregate Replication State of the RestorePoint from source to the target region as well as replication status of individual Disk Restore Points |
| disk.diskRestorePoint.id | String | A unique id for each Disk attached to the VM |
| completionPercent | String | Shows the completion percentage of cross-region copy/replication |