

---

# PowerShell-Module Documentation

*Release latest*

Sep 16, 2017



<b>1</b>	<b>Requirements</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Option 1: PowerShell Gallery (Recommended) . . . . .	5
2.2	Option 2: Installer Script . . . . .	5
2.3	Option 3: Manual Installation . . . . .	6
2.4	Verification . . . . .	6
<b>3</b>	<b>Getting Started</b>	<b>7</b>
3.1	Connecting to the Rubrik Cluster . . . . .	7
3.2	Commands and Help . . . . .	7
3.3	Gathering Data . . . . .	8
3.4	Modifying Data . . . . .	8
<b>4</b>	<b>Project Architecture</b>	<b>11</b>
<b>5</b>	<b>Support</b>	<b>13</b>
<b>6</b>	<b>Contribution</b>	<b>15</b>
6.1	Creating a New Function . . . . .	15
6.2	Storing API Data . . . . .	15
6.3	Building a Query . . . . .	16
6.4	Building a Body . . . . .	17
6.5	Building a Filter . . . . .	18
6.6	Updating the Module Manifest . . . . .	19
<b>7</b>	<b>Licensing</b>	<b>21</b>
<b>8</b>	<b>Frequently Asked Questions</b>	<b>23</b>
<b>9</b>	<b>Connect Commands</b>	<b>25</b>
9.1	Connect-Rubrik . . . . .	25
<b>10</b>	<b>Disconnect Commands</b>	<b>27</b>
10.1	Disconnect-Rubrik . . . . .	27
<b>11</b>	<b>Export Commands</b>	<b>29</b>
11.1	Export-RubrikDatabase . . . . .	29

11.2	Export-RubrikReport . . . . .	30
<b>12</b>	<b>Get Commands</b>	<b>33</b>
12.1	Get-RubrikAPIVersion . . . . .	33
12.2	Get-RubrikDatabase . . . . .	34
12.3	Get-RubrikDatabaseFiles . . . . .	35
12.4	Get-RubrikDatabaseMount . . . . .	36
12.5	Get-RubrikFileset . . . . .	37
12.6	Get-RubrikFilesetTemplate . . . . .	38
12.7	Get-RubrikHost . . . . .	39
12.8	Get-RubrikMount . . . . .	40
12.9	Get-RubrikReport . . . . .	41
12.10	Get-RubrikReportData . . . . .	41
12.11	Get-RubrikRequest . . . . .	42
12.12	Get-RubrikSLA . . . . .	43
12.13	Get-RubrikSnapshot . . . . .	44
12.14	Get-RubrikSoftwareVersion . . . . .	45
12.15	Get-RubrikUnmanagedObject . . . . .	45
12.16	Get-RubrikVersion . . . . .	46
12.17	Get-RubrikVM . . . . .	46
<b>13</b>	<b>Move Commands</b>	<b>49</b>
13.1	Move-RubrikMountVMDK . . . . .	49
<b>14</b>	<b>New Commands</b>	<b>51</b>
14.1	New-RubrikDatabaseMount . . . . .	51
14.2	New-RubrikHost . . . . .	52
14.3	New-RubrikMount . . . . .	53
14.4	New-RubrikReport . . . . .	54
14.5	New-RubrikSLA . . . . .	54
14.6	New-RubrikSnapshot . . . . .	55
<b>15</b>	<b>Protect Commands</b>	<b>57</b>
15.1	Protect-RubrikDatabase . . . . .	57
15.2	Protect-RubrikFileset . . . . .	58
15.3	Protect-RubrikTag . . . . .	59
15.4	Protect-RubrikVM . . . . .	60
<b>16</b>	<b>Remove Commands</b>	<b>63</b>
16.1	Remove-RubrikDatabaseMount . . . . .	63
16.2	Remove-RubrikFileset . . . . .	64
16.3	Remove-RubrikHost . . . . .	65
16.4	Remove-RubrikMount . . . . .	65
16.5	Remove-RubrikReport . . . . .	66
16.6	Remove-RubrikSLA . . . . .	67
16.7	Remove-RubrikUnmanagedObject . . . . .	67
<b>17</b>	<b>Restore Commands</b>	<b>69</b>
17.1	Restore-RubrikDatabase . . . . .	69
<b>18</b>	<b>Set Commands</b>	<b>71</b>
18.1	Set-RubrikBlackout . . . . .	71
18.2	Set-RubrikDatabase . . . . .	72
18.3	Set-RubrikMount . . . . .	72
18.4	Set-RubrikVM . . . . .	73

<b>19 Sync Commands</b>	<b>75</b>
19.1 Sync-RubrikAnnotation . . . . .	75
19.2 Sync-RubrikTag . . . . .	76
<b>20 Auditing Workflows</b>	<b>77</b>
20.1 Failed Tasks . . . . .	77
<b>21 Backup Validation Workflows</b>	<b>79</b>



This is a community project that provides a Windows PowerShell module for managing and monitoring Rubrik's Converged Data Management fabric by way of published RESTful APIs. If you're looking to perform interactive automation, setting up scheduled tasks, leverage an orchestration engine, or need ad-hoc operations, this module is intended to be valuable to your needs. The code is open source, and [available on GitHub](#).

Below is a quick YouTube video that explains how to begin using the module.





# CHAPTER 1

---

## Requirements

---

The code assumes that you've already deployed at least one Rubrik Brik into your environment and have completed the initial configuration process to form a cluster. At a minimum, make sure you have installed the following:

1. PowerShell version 4 or higher
2. PowerCLI version 6.0 or higher
3. Rubrik version 2.2 or higher
4. (optional) Windows Management Framework 5.0

Note: Although an alpha build of PowerShell exists for Linux and Mac OS environments, the code is currently being written and validated using a Microsoft Windows Server environment.



This repository contains a folder named **Rubrik**. The folder needs to be installed into one of your PowerShell Module Paths using one of the installation methods outlined in the next section. To see the full list of available PowerShell Module paths, use `$env:PSModulePath.split(';')` in a PowerShell console.

Common PowerShell module paths include:

1. **Current User:** `%USERPROFILE%\Documents\WindowsPowerShell\Modules\`
2. **All Users:** `%ProgramFiles%\WindowsPowerShell\Modules\`
3. **OneDrive:** `$env:OneDrive\Documents\WindowsPowerShell\Modules\`

### Option 1: PowerShell Gallery (Recommended)

1. Ensure you have the [Windows Management Framework 5.0](#) or greater installed.
2. Open a Powershell console with the *Run as Administrator* option.
3. Run `Set-ExecutionPolicy` using the parameter *RemoteSigned* or *Bypass*.
4. Run `Install-Module -Name Rubrik -Scope CurrentUser` to download the module from the PowerShell Gallery. Note that the first time you install from the remote repository it may ask you to first trust the repository.

### Option 2: Installer Script

1. Download the [master branch](#) to your workstation.
2. Open a Powershell console with the *Run as Administrator* option.
3. Run `Set-ExecutionPolicy` using the parameter *RemoteSigned* or *Bypass*.
4. Run the `Install-Rubrik.ps1` script in the root of this repository and follow the prompts to install, upgrade, or delete your Rubrik Module contents.

## Option 3: Manual Installation

1. Download the [master branch](#) to your workstation.
2. Copy the contents of the *Rubrik* folder onto your workstation into the desired PowerShell Module path.
3. Open a Powershell console with the *Run as Administrator* option.
4. Run `Set-ExecutionPolicy` using the parameter *RemoteSigned* or *Bypass*.

Once installation is complete, you can validate that the module exists by running `Get-Module -ListAvailable Rubrik`.

## Verification

PowerShell will create a folder for each new version of any module you install. It's a good idea to check and see what version(s) you have installed and running in the session. To begin, let's see what versions of the Rubrik Module are installed:

```
Get-Module -ListAvailable Rubrik
```

The `-ListAvailable` switch will pull up all installed versions from any path found in `$env:PSModulePath`. Check to make sure the version you wanted is installed. You can safely remove old versions, if desired.

To see which version is currently loaded, use:

```
Get-Module Rubrik
```

If nothing is returned, you need to first load the module by using:

```
Import-Module Rubrik
```

If you wish to load a specific version, use:

```
Import-Module Rubrik -RequiredVersion #.#.#.#
```

Where “#.#.#.#” represents the version number.

Now that you have the Rubrik module installed on your workstation, there's a few beginner commands that you can explore to feel comfortable with the available functions.

### Connecting to the Rubrik Cluster

To begin, let's connect to a Rubrik cluster. To keep things simple, we'll do the first command without any supplied parameters.

- Open a PowerShell session
- Type `Connect-Rubrik` and press enter.

A prompt will appear asking you for a server. Enter the IP address or fully qualified domain name (FQDN) of any node in the cluster. An additional prompt will appear asking for your user credentials. Once entered, you will see details about the newly created connection.

At this point, you are authenticated and ready to begin issuing commands to the cluster.

### Commands and Help

What if we didn't know that `Connect-Rubrik` exists? To see a list of all available commands, type in `Get-Command -Module Rubrik`. This will display a list of every function available in the module. Note that all commands are in the format of **Verb-RubrikNoun**. This has two benefits:

- Adheres to the Microsoft requirements for PowerShell functions.
- Use of "Rubrik" in the name avoids collisions with anyone else's commands.

For details on a command, use the PowerShell help command `Get-Help`. For example, to get help on the `Connect-Rubrik` function, use the following command:

```
Get-Help Connect-Rubrik
```

This will display a description about the command. For details and examples, use the `-Full` parameter on the end.

```
Get-Help Connect-Rubrik -Full
```

## Gathering Data

Let's get information on the cluster. The use of any command beginning with the word **get** is safe to use. No data will be modified. These are good commands to use if this is your first time using PowerShell.

We'll start by looking up the version running on the Rubrik cluster. Enter the command below and press enter:

```
Get-RubrikVersion
```

The result is fairly simple: the command will output the cluster's code version. How about something a bit more complex? Try getting all of the SLA Domain details from the cluster. Here's the command:

```
Get-RubrikSLA
```

Lots of stuff should be scrolling across the screen. You're seeing details on every SLA Domain known by the cluster at a very detailed level. If you want to see just one SLA Domain, tell the command to limit results. You can do this by using a *parameter*. Parameters are ways to control a function. Try it now.

```
Get-RubrikSLA -SLA "Gold"
```

The `-SLA` portion is a parameter and "Gold" is a value for the parameter. This effectively asks the function to limit results to one SLA Domain: gold. Easy, right?

For a full list of available parameters and examples, use `Get-Help Get-RubrikSLA -Full`. Every Rubrik command has native help available.

## Modifying Data

Not every command will be about gathering data. Sometimes you'll want to modify or delete data, too. The process is nearly the same, although some safeguards have been implemented to protect against errant modifications. Let's start with an easy one.

This works best if you have a test virtual machine that you don't care about. Make sure that virtual machine is visible to the Rubrik cluster. To validate this, use the following command:

```
Get-RubrikVM -VM "Name"
```

Make sure to replace "Name" with the actual name of the virtual machine. If you received data back from Rubrik, you can be sure that this virtual machine is known to the cluster and can be modified.

**Note:** The double quotes are required if your virtual machine has any spaces in the name. It's generally considered a good habit to always use double quotes around the name of objects.

Let's protect this virtual machine with the "Gold" SLA Domain. To do this, use the following command:

```
Protect-RubrikVM -VM "Name" -SLA "Gold"
```

Before the change is made, a prompt will appear asking you to confirm the change.

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Assign SLA Domain 1234567890" on target "Name".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y
↵") :
```

This is a safeguard. You can either take the default action of “Yes” by pressing enter, or type “N” if you entered the wrong name or changed your mind. If you want to skip the confirmation check all together, use the `-Confirm:$false` parameter like this:

```
Protect-RubrikVM -VM "Name" -SLA "Gold" -Confirm:$false
```

This will make the change without asking for confirmation. Be careful!





---

## Project Architecture

---

This page contains details on the artifacts found within the repository.

- docs: Documentation on the module
- templates: Templates for creating your own functions
- tests: Pester unit tests used to validate the public functions
- workflows: Sample workflows for more complex automation tasks
- **Rubrik: The parent folder containing the module**
  - Private: Private functions that are used internally by the module
  - Public: Published functions that are available to the PowerShell session
  - Rubrik.psd1: Module manifest
  - Rubrik.psm1: Script module file



## CHAPTER 5

---

### Support

---

The community module is not officially supported and should be **used at your own risk**.

A future release will offer API versioning and may also include formal support.

To report a bug, request an enhancement, or provide feedback about this project, please [open an issue](#).



Everyone is welcome to contribute to this project. Here are the high level steps involved:

1. Create a fork of the project into your own repository.
2. From your fork, create a new feature branch (other than master) that expresses your feature or enhancement.
3. Make all your necessary changes in your feature branch.
4. Create a pull request with a description on what was added or removed and details explaining the changes in lines of code.

If approved, project owners will merge it.

## Creating a New Function

To keep the module simple, each function is saved as its own ps1 file and stored in `Rubrik\Public`. The naming format is `Verb-RubrikNoun.ps1` in which `Verb` is a Microsoft approved verb (Use `Get-Verb` to see the list) and the noun is singular. Therefore “`Get-RubrikVM`” is acceptable but “`Get-RubrikVMs`” is not. You can quickly bootstrap this process by using either the `template\Verb-RubrikNoun.ps1` file as a template or pick from an existing function that contains code you’d like to re-use.

Every function needs to have two specific variables included and should not be removed:

1. `[String]$Server = $global:RubrikConnection.server` - Stores the Rubrik cluster’s IP or FQDN
2. `[String]$api = $global:RubrikConnection.api` - Stores the API version

## Storing API Data

All API specific data for a function is abstracted into a helper function located in `Rubrik\Private` named `Get-RubrikAPIData.ps1`. This separates function logic from function data and allows for simple iterations to cross different API versions while also making unit testing much simpler. If your function is going to call the API

directly, all API specific requirements should be stored in `Get-RubrikAPIData.ps1`. Data is stored in a hashtable like the sample below:

```
Example = @{
  v1 = @{
    Description = 'Details about the API endpoint'
    URI         = 'The URI expressed as /api/v#/endpoint'
    Method      = 'Method to use against the endpoint'
    Body        = 'Parameters to use in the body'
    Query       = 'Parameters to use in the URI query'
    Result      = 'If the result content is stored in a higher level key, express it,
↳here to be unwrapped in the return'
    Filter      = 'If the result content needs to be filtered based on key names,
↳express them here'
    Success     = 'The expected HTTP status code for a successful call'
  }
}
```

Once data for the endpoint is stored, the function template should work in most cases. Most of the heavy lifting is handled by private functions that are called in order to build the query (path) parameters, construct a body payload, and filter the results.

## Building a Query

For many endpoints, a query is constructed using the endpoint's path. This is fairly common when using the `id` value.

Below, let's look at `Get-RubrikDatabase` and how it builds a query. There are 4 different parameters available for the query: `instance_id`, `effective_sla_domain_id`, `primary_cluster_id`, and `is_relic`. By placing these objects in the `Query` section, we've informed the function that they are available.

Note that the key and value match because this is the first version of the API; should the parameter name change in the future, the value would change to match it, but the key would remain static to avoid re-writing anything in the function itself.

```
'Get-RubrikDatabase' = @{
  v1 = @{
    Description = 'Returns a list of summary information for Microsoft SQL databases.'
    URI         = '/api/v1/mssql/db'
    Method      = 'Get'
    Body        = ''
    Query       = @{
      instance_id           = 'instance_id'
      effective_sla_domain_id = 'effective_sla_domain_id'
      primary_cluster_id   = 'primary_cluster_id'
      is_relic              = 'is_relic'
    }
    Result      = 'data'
    Filter      = @{
      'Name' = 'name'
      'SLA'  = 'effectiveSlaDomainName'
      'Hostname' = 'rootProperties.rootName'
      'Instance' = 'instanceName'
    }
    Success     = '200'
  }
}
```

The parameter names aren't very user friendly. In order to use friendly parameter names within the function, a relationship is created using an alias. Below are the parameters available in the `Get-RubrikDatabase` function. Notice how the `[Switch]$Relic` switch has an alias of `[Alias('is_relic')]` to match the API's parameter? This creates a relationship so that the function knows to build a query using the value of this parameter.

In this case, if the user sets the switch to `$true`, the `is_relic` query will be added to the path. The same goes for `$PrimaryClusterID` and `$SLAID`.

```
Param(
    # Name of the database
    [Alias('Database')]
    [String]$Name,
    # Filter results to include only relic (removed) databases
    [Alias('is_relic')]
    [Switch]$Relic,
    # SLA Domain policy assigned to the database
    [String]$SLA,
    # Name of the database instance
    [String]$Instance,
    # Name of the database host
    [String]$Hostname,
    # Filter the summary information based on the primarycluster_id of the primary_
↪Rubrik cluster. Use *_local* as the primary_cluster_id of the Rubrik cluster that_
↪is hosting the current REST API session.
    [Alias('primary_cluster_id')]
    [String]$PrimaryClusterID,
    # Rubrik's database id value
    [Parameter(ValueFromPipelineByPropertyName = $true)]
    [String]$id,
    # SLA id value
    [Alias('effective_sla_domain_id')]
    [String]$SLAID,
    # Rubrik server IP or FQDN
    [String]$Server = $global:RubrikConnection.server,
    # API version
    [ValidateNotNullorEmpty()]
    [String]$api = $global:RubrikConnection.api
)
```

If the Query building function doesn't find a particular parameter in the API data, it ignores the value with respect to building a query.

## Building a Body

Constructing a body payload is very similar to a query. Let's look at the `New-RubrikMount` function as an example. Notice how it has a body section with parameters defined? The body parameters follow the same rules as query parameters do: include the key/value pairs in the API data, and then use aliases within the function to build a relationship.

```
'New-RubrikMount' = @{
    v1 = @{
        Description = 'Create a live mount request with given configuration'
        URI         = '/api/v1/vmware/vm/snapshot/{id}/mount'
        Method      = 'Post'
        Body        = @{
            hostId = 'hostId'
```

```

    vmName           = 'vmName'
    dataStoreName    = 'dataStoreName'
    disableNetwork   = 'disableNetwork'
    removeNetworkDevices = 'removeNetworkDevices'
    powerOn          = 'powerOn'
  }
  Query             = ''
  Result            = ''
  Filter            = ''
  Success           = '202'
}
}

```

And here's the PowerShell code to see the body parameter aliases. See how `[String]$MountName` has an alias of `[Alias('vmName')]` to avoid user confusion? And because that value is declared in the body section of the API data, the private functions know to use that parameter to construct the body payload.

```

Param(
  # Rubrik id of the snapshot
  [Parameter(Mandatory = $true, ValueFromPipelineByPropertyName = $true)]
  [String]$id,
  # ID of host for the mount to use
  [String]$HostID,
  # Name of the mounted VM
  [Alias('vmName')]
  [String]$MountName,
  # Name of the data store to use/create on the host
  [String]$DatastoreName,
  # Whether the network should be disabled on mount. This should be set true to
  → avoid ip conflict in case of static IPs.
  [Switch]$DisableNetwork,
  # Whether the network devices should be removed on mount.
  [Switch]$RemoveNetworkDevices,
  # Whether the VM should be powered on after mount.
  [Alias('powerOn')]
  [Switch]$PowerState,
  # Rubrik server IP or FQDN
  [String]$Server = $global:RubrikConnection.server,
  # API version
  [String]$api = $global:RubrikConnection.api
)

```

## Building a Filter

Not every API endpoint has the ability to filter results as desired. In those cases, the filter section of the API data is used to provide additional result filtering for the user.

Let's take a peek at `Get-RubrikVM` as an example. Notice how the filter section is different from the query and body sections. The filter keys correspond to the function's actual parameter names. The values correspond to the keys found in the result data. This relationship is used to filter specific key/value pairs in the result for user driven filter criteria.

```

'Get-RubrikVM' = @{
  v1 = @{
    Description = 'Get summary of all the VMs'
  }
}

```



```

URI          = '/api/v1/vmware/vm'
Method       = 'Get'
Body        = ''
Query       = @{
    is_relic           = 'is_relic'
    name               = 'name'
    effective_sla_domain_id = 'effective_sla_domain_id'
}
Result      = 'data'
Filter     = @{
    'Name' = 'name'
    'SLA'  = 'effectiveSlaDomainName'
}
Success    = '200'
}
}

```

Let's take the 'SLA' = 'effectiveSlaDomainName' as an example: a user enters a friendly SLA Domain name into the \$SLA parameter. When the results come back from the endpoint, this friendly name is compared against the results. PowerShell looks at the effectiveSlaDomainName key and filters out anything that doesn't match. If the user entered the word "Gold", the function would filter out any results that aren't part of the "Gold" SLA Domain.

There is no need to create an alias because the actual parameter name is used (without the \$ symbol).

```

Param(
    # Name of the virtual machine
    [Parameter(Position = 0, ValueFromPipelineByPropertyName = $true)]
    [Alias('VM')]
    [String]$Name,
    # Filter results to include only relic (removed) virtual machines
    [Alias('is_relic')]
    [Switch]$Relic,
    # SLA Domain policy assigned to the virtual machine
    [String]$SLA,
    # Virtual machine id
    [String]$id,
    # SLA id value
    [Alias('effective_sla_domain_id')]
    [String]$SLAID,
    # Rubrik server IP or FQDN
    [String]$Server = $global:RubrikConnection.server,
    # API version
    [String]$api = $global:RubrikConnection.api
)

```

## Updating the Module Manifest

The final step is to update the module manifest and add the new function to the FunctionsToExport value. This is done in the Rubrik\Rubrik.psd1 file.



## CHAPTER 7

---

### Licensing

---

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



## CHAPTER 8

---

### Frequently Asked Questions

---

This section will contain a list of questions that have been received (and answered) by the Project Team.



---

## Connect Commands

---

This page contains details on **Connect** commands.

### Connect-Rubrik

**NAME** Connect-Rubrik

**SYNOPSIS** Connects to Rubrik and retrieves a token value for authentication

**SYNTAX** Connect-Rubrik [-Server] <String> [[-Username] <String>] [[-Password] <SecureString>] [[-Credential] <Object>] [<CommonParameters>]

**DESCRIPTION** The Connect-Rubrik function is used to connect to the Rubrik RESTful API and supply credentials to the /login method. Rubrik then returns a unique token to represent the user's credentials for subsequent calls. Acquire a token before running other Rubrik cmdlets. Note that you can pass a username and password or an entire set of credentials.

#### PARAMETERS

**-Server <String>** The IP or FQDN of any available Rubrik node within the cluster

**-Username <String>** Username with permissions to connect to the Rubrik cluster Optionally, use the Credential parameter

**-Password <SecureString>** Password for the Username provided Optionally, use the Credential parameter

**-Credential <Object>** Credentials with permission to connect to the Rubrik cluster Optionally, use the Username and Password parameters

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

---

#### EXAMPLE 1

---

```
PS C:>Connect-Rubrik -Server 192.168.1.1 -Username admin
```

This will connect to Rubrik with a username of “admin” to the IP address 192.168.1.1. The prompt will request a secure password.

————— EXAMPLE 2 —————

```
PS C:>Connect-Rubrik -Server 192.168.1.1 -Username admin -Password (ConvertTo-SecureString “secret” -asplaintext -force)
```

If you need to pass the password value in the cmdlet directly, use the ConvertTo-SecureString function.

————— EXAMPLE 3 —————

```
PS C:>Connect-Rubrik -Server 192.168.1.1 -Credential (Get-Credential)
```

Rather than passing a username and secure password, you can also opt to submit an entire set of credentials using the -Credentials parameter.

**REMARKS** To see the examples, type: “get-help Connect-Rubrik -examples”. For more information, type: “get-help Connect-Rubrik -detailed”. For technical information, type: “get-help Connect-Rubrik -full”. For online help, type: “get-help Connect-Rubrik -online”



---

## Disconnect Commands

---

This page contains details on **Disconnect** commands.

### Disconnect-Rubrik

**NAME** Disconnect-Rubrik

**SYNOPSIS** Disconnects from a Rubrik cluster

**SYNTAX** Disconnect-Rubrik [[-id] <String>] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Disconnect-Rubrik function is used to disconnect from a Rubrik cluster. This is done by supplying the bearer token and requesting that the session be deleted.

#### PARAMETERS

**-id <String>** Session id

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

---

#### EXAMPLE 1

```
PS C:>Disconnect-Rubrik -Confirm:$false
```

This will close the current session and invalidate the current session token without prompting for confirmation

---

#### EXAMPLE 2

```
PS C:>$rubrikConnection = $RubrikConnections[1]
```

Disconnect-Rubrik This will close the second session and invalidate the second session token Note: The \$rubrikConnections variable holds session details on all established sessions

The \$rubrikConnection variable holds the current, active session If you wish to change sessions, simply update the value of \$rubrikConnection to another session held within \$rubrikConnections

**REMARKS** To see the examples, type: “get-help Disconnect-Rubrik -examples”. For more information, type: “get-help Disconnect-Rubrik -detailed”. For technical information, type: “get-help Disconnect-Rubrik -full”. For online help, type: “get-help Disconnect-Rubrik -online”

---

## Export Commands

---

This page contains details on **Export** commands.

### Export-RubrikDatabase

**NAME** Export-RubrikDatabase

**SYNOPSIS** Connects to Rubrik exports a database to a MSSQL instance

**SYNTAX** Export-RubrikDatabase -Id <String> [-MaxDataStreams <Int32>] [-TimestampMs <Int64>] [-FinishRecovery] [-TargetInstanceId <String>] [-TargetDatabaseName <String>] [-Server <String>] [-api <String>] [-TargetFilePaths <PSObject[]>] [-WhatIf] [-Confirm] [<CommonParameters>]

Export-RubrikDatabase -Id <String> [-MaxDataStreams <Int32>] [-RecoveryDateTime <DateTime>] [-FinishRecovery] [-TargetInstanceId <String>] [-TargetDatabaseName <String>] [-Server <String>] [-api <String>] [-TargetFilePaths <PSObject[]>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Export-RubrikDatabase command will request a database export from a Rubrik Cluster to a MSSQL instance

#### PARAMETERS

- Id <String> Rubrik identifier of database to be exported
- MaxDataStreams <Int32> Number of parallel streams to copy data
- TimestampMs <Int64> Recovery Point desired in the form of Epoch with Milliseconds
- RecoveryDateTime <DateTime> Recovery Point desired in the form of DateTime value
- FinishRecovery [<SwitchParameter>] Take database out of recovery mode after export
- TargetInstanceId <String> Rubrik identifier of MSSQL instance to export to
- TargetDatabaseName <String> Name to give database upon export
- Server <String> Rubrik server IP or FQDN

**-api** <String> API version

**-TargetFilePaths** <PSObject[]> Optional Export File Hash table Array

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Export-RubrikDatabase -id MssqlDatabase:::c5ecf3ef-248d-4bb2-8fe1-4d3c820a0e38 -targetInstanceId
MssqlInstance:::0085b247-e718-4177-869f-e3ae1f7bb503 -targetDatabaseName ReportServer -finishRecovery
-maxDataStreams 4 -timestampMs 1492661627000
```

————— EXAMPLE 2 —————

```
PS C:>Export-RubrikDatabase -id $db.id -recoveryDateTime (Get-Date (Get-RubrikDatabase
$db).latestRecoveryPoint) -targetInstanceId $db2.instanceId -targetDatabaseName 'BAR_EXP' -
targetFilePaths $targetfiles -maxDataStreams 1
```

Restore the \$db (where \$db is the output of a Get-RubrikDatabase call) to the most recent recovery point for that database. New file paths are in the \$targetfiles array:

```
$targetfiles += @{logicalName='BAR_1';exportPath='E:SQLFilesDataBAREXP'} $targetfiles +=
@{logicalName='BAR_LOG';exportPath='E:SQLFilesLogBAREXP' }
```

**REMARKS** To see the examples, type: “get-help Export-RubrikDatabase -examples”. For more information, type: “get-help Export-RubrikDatabase -detailed”. For technical information, type: “get-help Export-RubrikDatabase -full”. For online help, type: “get-help Export-RubrikDatabase -online”

## Export-RubrikReport

**NAME** Export-RubrikReport

**SYNOPSIS** Retrieves link to a CSV file for a Rubrik Envision report

**SYNTAX** Export-RubrikReport [-id] <String> [[-TimezoneOffset] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Export-RubrikReport cmdlet is used to pull the link to a CSV file for a Rubrik Envision report

### PARAMETERS

**-id** <String> ID of the report.

**-TimezoneOffset** <String> Timezone offset from UTC in minutes.

**-Server** <String> Rubrik server IP or FQDN

**-api** <String> API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Export-RubrikReport -id '11111111-2222-3333-4444-555555555555' -timezone_offset 120
```

This will return the link to a CSV file for report id “11111111-2222-3333-4444-555555555555”

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikReport -Name 'Protection Tasks Details' | Export-RubrikReport
```

This will return the link to a CSV file for report named “Protection Tasks Details”

**REMARKS** To see the examples, type: “get-help Export-RubrikReport -examples”. For more information, type: “get-help Export-RubrikReport -detailed”. For technical information, type: “get-help Export-RubrikReport -full”. For online help, type: “get-help Export-RubrikReport -online”



This page contains details on **Get** commands.

## Get-RubrikAPIVersion

**NAME** Get-RubrikAPIVersion

**SYNOPSIS** Connects to Rubrik and retrieves the current API version

**SYNTAX** Get-RubrikAPIVersion [-Server] <String> [[-id] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikVersion cmdlet will retrieve the API version actively running on the system. This does not require authentication.

### PARAMETERS

**-Server <String>** Rubrik server IP or FQDN

**-id <String>** ID of the Rubrik cluster or me for self

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

---

#### EXAMPLE 1

```
PS C:>Get-RubrikAPIVersion -Server 192.168.1.100
```

This will return the running API version on the Rubrik cluster reachable at the address 192.168.1.100

**REMARKS** To see the examples, type: “get-help Get-RubrikAPIVersion -examples”. For more information, type: “get-help Get-RubrikAPIVersion -detailed”. For technical information, type: “get-help Get-RubrikAPIVersion -full”. For online help, type: “get-help Get-RubrikAPIVersion -online”

## Get-RubrikDatabase

**NAME** Get-RubrikDatabase

**SYNOPSIS** Retrieves details on one or more databases known to a Rubrik cluster

**SYNTAX** Get-RubrikDatabase [[-Name] <String>] [-Relic] [[-SLA] <String>] [[-Instance] <String>] [[-Hostname] <String>] [[-PrimaryClusterID] <String>] [[-id] <String>] [[-SLAID] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikDatabase cmdlet is used to pull a detailed data set from a Rubrik cluster on any number of databases. To narrow down the results, use the host and instance parameters to limit your search to a smaller group of objects. Alternatively, supply the Rubrik database ID to return only one specific database.

### PARAMETERS

**-Name <String>** Name of the database

**-Relic [<SwitchParameter>]** Filter results to include only relic (removed) databases

**-SLA <String>** SLA Domain policy assigned to the database

**-Instance <String>** Name of the database instance

**-Hostname <String>** Name of the database host

**-PrimaryClusterID <String>** Filter the summary information based on the primarycluster\_id of the primary Rubrik cluster. Use **\_local** as the primary\_cluster\_id of the Rubrik cluster that is hosting the current REST API session.

**-id <String>** Rubrik's database id value

**-SLAID <String>** SLA id value

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

#### EXAMPLE 1

```
PS C:>Get-RubrikDatabase -Name 'DB1' -SLA Gold
```

This will return details on all databases named DB1 protected by the Gold SLA Domain on any known host or instance.

#### EXAMPLE 2

```
PS C:>Get-RubrikDatabase -Name 'DB1' -Host 'Host1' -Instance 'MSSQLSERVER'
```

This will return details on a database named "DB1" living on an instance named "MSSQLSERVER" on the host named "Host1".

#### EXAMPLE 3

```
PS C:>Get-RubrikDatabase -Relic
```

This will return all removed databases that were formerly protected by Rubrik.

#### EXAMPLE 4

```
PS C:>Get-RubrikDatabase -id 'MssqlDatabase:::aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee'
```



This will return details on a single database matching the Rubrik ID of “MssqlDatabase:::aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee” Note that the database ID is globally unique and is often handy to know if tracking a specific database for longer workflows, whereas some values are not unique (such as nearly all hosts having one or more databases named “model”) and more difficult to track by name.

**REMARKS** To see the examples, type: “get-help Get-RubrikDatabase -examples”. For more information, type: “get-help Get-RubrikDatabase -detailed”. For technical information, type: “get-help Get-RubrikDatabase -full”. For online help, type: “get-help Get-RubrikDatabase -online”

## Get-RubrikDatabaseFiles

**NAME** Get-RubrikDatabaseFiles

**SYNOPSIS** Connects to Rubrik and retrieves all the data files for a SQL Server Database snapshot

**SYNTAX** Get-RubrikDatabaseFiles [[-Id] <String>] [[-RecoveryDateTime] <DateTime>] [[-time] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikDatabaseFiles cmdlet will return all the available databasem files for a database snapshot. This is based on the recovery time for the database, as file locations could change between snapshots and log backups. If no date time is provided, the database’s latest recovery point will be used

**\*WARNING\*** This is based on an internal endpoint and is subject to change by the REST API team.

### PARAMETERS

**-Id <String>** Rubrik’s id of the mount

**-RecoveryDateTime <DateTime>** Recovery Point desired in the form of DateTime value

**-time <String>** Recovery Point desired in the form of a UTC string (yyyy-MM-ddTHH:mm:ss)

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikDatabaseFiles -id '11111111-2222-3333-4444-555555555555'
```

This will return files for database id “11111111-2222-3333-4444-555555555555”.

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikDatabaseMount -id '11111111-2222-3333-4444-555555555555' -RecoveryDateTime (Get-Date).AddDays(-1)
```

This will return details on mount id “11111111-2222-3333-4444-555555555555” from a recovery point one day ago, assuming that recovery point exists.

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikDatabaseMount -id '11111111-2222-3333-4444-555555555555' -time '2017-08-08T01:15:00Z'
```

This will return details on mount id “11111111-2222-3333-4444-555555555555” from UTC ‘2017-08-08 01:15:00’, assuming that recovery point exists.

**REMARKS** To see the examples, type: “get-help Get-RubrikDatabaseFiles -examples”. For more information, type: “get-help Get-RubrikDatabaseFiles -detailed”. For technical information, type: “get-help Get-RubrikDatabaseFiles -full”. For online help, type: “get-help Get-RubrikDatabaseFiles -online”

## Get-RubrikDatabaseMount

**NAME** Get-RubrikDatabaseMount

**SYNOPSIS** Connects to Rubrik and retrieves details on mounts for a SQL Server Database

**SYNTAX** Get-RubrikDatabaseMount [[-Id] <String>] [[-SourceDatabaseId] <String>] [[-SourceDatabaseName] <String>] [[-TargetInstanceId] <String>] [[-MountedDatabaseName] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikMount cmdlet will accept one of several different query parameters and retrieve the database Live Mount information for that criteria.

### PARAMETERS

**-Id <String>** Rubrik’s id of the mount

**-SourceDatabaseId <String>** Filters live mounts by database source id

**-SourceDatabaseName <String>** Filters live mounts by database source name

**-TargetInstanceId <String>** Filters live mounts by database source name

**-MountedDatabaseName <String>** Filters live mounts by database source name

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikDatabaseMount
```

This will return details on all mounted databases.

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikDatabaseMount -id '11111111-2222-3333-4444-555555555555'
```

This will return details on mount id “11111111-2222-3333-4444-555555555555”.

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikDatabaseMount -source_database_id (Get-RubrikDatabase -HostName FOO -Instance MSSQLSERVER -Database BAR).id
```

This will return details for any mounts found using the id value from a database named BAR on the FOO default instance.

————— EXAMPLE 4 —————

```
PS C:>Get-RubrikDatabaseMount -source_database_name BAR
```

This returns any mounts where the source database is named BAR.

————— EXAMPLE 5 —————

```
PS C:>Get-RubrikDatabaseMount -mounted_database_name BAR_LM
```

This returns any mounts with the name BAR\_LM

**REMARKS** To see the examples, type: “get-help Get-RubrikDatabaseMount -examples”. For more information, type: “get-help Get-RubrikDatabaseMount -detailed”. For technical information, type: “get-help Get-RubrikDatabaseMount -full”. For online help, type: “get-help Get-RubrikDatabaseMount -online”

## Get-RubrikFileset

**NAME** Get-RubrikFileset

**SYNOPSIS** Retrieves details on one or more filesets known to a Rubrik cluster

**SYNTAX** Get-RubrikFileset [[-Name] <String>] [-Relic] [[-SLA] <String>] [[-HostName] <String>] [[-TemplateID] <String>] [[-PrimaryClusterID] <String>] [[-id] <String>] [[-SLAID] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikFileset cmdlet is used to pull a detailed data set from a Rubrik cluster on any number of filesets A number of parameters exist to help narrow down the specific fileset desired Note that a fileset name is not required; you can use params (such as HostName and SLA) to do lookup matching filesets

### PARAMETERS

**-Name <String>** Name of the fileset

**-Relic [<SwitchParameter>]** Filter results to include only relic (removed) filesets

**-SLA <String>** SLA Domain policy assigned to the database

**-HostName <String>** Name of the host using a fileset

**-TemplateID <String>** Filter the summary information based on the ID of a fileset template.

**-PrimaryClusterID <String>** Filter the summary information based on the primarycluster\_id of the primary Rubrik cluster. Use **\_local** as the primary\_cluster\_id of the Rubrik cluster that is hosting the current REST API session.

**-id <String>** Rubrik’s fileset id

**-SLAID <String>** SLA id value

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikFileset -Name ‘C_Drive’
```

This will return details on the fileset named “C\_Drive” assigned to any hosts

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikFileset -Name ‘C_Drive’ -HostName ‘Server1’
```

This will return details on the fileset named “C\_Drive” assigned to only the “Server1” host

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikFileset -Name ‘C_Drive’ -SLA Gold
```

This will return details on the fileset named “C\_Drive” assigned to any hosts with an SLA Domain matching “Gold”

————— EXAMPLE 4 —————

```
PS C:>Get-RubrikFileset -id 'Fileset:::111111-2222-3333-4444-555555555555'
```

This will return the filset matching the Rubrik global id value of “Fileset:::111111-2222-3333-4444-555555555555”

————— EXAMPLE 5 —————

```
PS C:>Get-RubrikFileset -Relic
```

This will return all removed filesets that were formerly protected by Rubrik.

**REMARKS** To see the examples, type: “get-help Get-RubrikFileset -examples”. For more information, type: “get-help Get-RubrikFileset -detailed”. For technical information, type: “get-help Get-RubrikFileset -full”. For online help, type: “get-help Get-RubrikFileset -online”

## Get-RubrikFilesetTemplate

**NAME** Get-RubrikFilesetTemplate

**SYNOPSIS** Retrieves details on one or more fileset templates known to a Rubrik cluster

**SYNTAX** Get-RubrikFilesetTemplate [[-Name] <String>] [[-OperatingSystemType] <String>] [[-PrimaryClusterID] <String>] [[-id] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikFilesetTemplate cmdlet is used to pull a detailed data set from a Rubrik cluster on any number of fileset templates

### PARAMETERS

**-Name <String>** Retrieve fileset templates with a name matching the provided name. The search is performed as a case-insensitive infix search.

**-OperatingSystemType <String>** Filter the summary information based on the operating system type of the fileset. Accepted values: ‘Windows’, ‘Linux’

**-PrimaryClusterID <String>** Filter the summary information based on the primarycluster\_id of the primary Rubrik cluster. Use **\_local** as the primary\_cluster\_id of the Rubrik cluster that is hosting the current REST API session.

**-id <String>** The ID of the fileset template

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikFilesetTemplate -Name 'Template1'
```

This will return details on all fileset templates named “Template1”

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikFilesetTemplate -OperatingSystemType 'Linux'
```

This will return details on all fileset templates that can be used against a Linux operating system type

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikFilesetTemplate -id '11111111-2222-3333-4444-555555555555'
```

This will return details on the fileset template matching id “11111111-2222-3333-4444-555555555555”

**REMARKS** To see the examples, type: “get-help Get-RubrikFilesetTemplate -examples”. For more information, type: “get-help Get-RubrikFilesetTemplate -detailed”. For technical information, type: “get-help Get-RubrikFilesetTemplate -full”. For online help, type: “get-help Get-RubrikFilesetTemplate -online”

## Get-RubrikHost

**NAME** Get-RubrikHost

**SYNOPSIS** Retrieve summary information for all hosts that are registered with a Rubrik cluster.

**SYNTAX** Get-RubrikHost [[-Name] <String>] [[-Type] <String>] [[-PrimaryClusterID] <String>] [[-id] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikHost cmdlet is used to retrieve information on one or more hosts that are being protected with the Rubrik Backup Service or directly as with the case of NAS shares.

### PARAMETERS

**-Name <String>** Retrieve hosts with a host name matching the provided name. The search type is infix

**-Type <String>** Filter the summary information based on the operating system type. Accepted values are ‘Windows’, ‘Linux’, ‘ANY’, ‘NONE’. Use NONE to only return information for hosts templates that do not have operating system type set. Use ANY to only return information for hosts that have operating system type set.

**-PrimaryClusterID <String>** Filter the summary information based on the primarycluster\_id of the primary Rubrik cluster. Use **\_local** as the primary\_cluster\_id of the Rubrik cluster that is hosting the current REST API session.

**-id <String>** ID of the registered host

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikHost
```

This will return all known hosts

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikHost -Hostname 'Server1'
```

This will return details on any hostname matching “Server1”

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikHost -Type 'Windows' -PrimaryClusterID 'local'
```

This will return details on all Windows hosts that are being protected by the local Rubrik cluster

EXAMPLE 4

```
PS C:>Get-RubrikHost -id 'Host:::111111-2222-3333-4444-555555555555'
```

This will return details specifically for the host id matching “Host:::111111-2222-3333-4444-555555555555”

**REMARKS** To see the examples, type: “get-help Get-RubrikHost -examples”. For more information, type: “get-help Get-RubrikHost -detailed”. For technical information, type: “get-help Get-RubrikHost -full”. For online help, type: “get-help Get-RubrikHost -online”

## Get-RubrikMount

**NAME** Get-RubrikMount

**SYNOPSIS** Connects to Rubrik and retrieves details on mounts for a VM

**SYNTAX** Get-RubrikMount [[-id] <String>] [[-VMID] <String>] [[-Server] <String>] [[-api] <String>] [<Common-Parameters>]

**DESCRIPTION** The Get-RubrikMount cmdlet will accept a VM id and return details on any mount operations that are active within Rubrik Due to the nature of names not being unique Note that this function requires the VM ID value, not the name of the virtual machine, since virtual machine names are not unique. It is suggested that you first use Get-RubrikVM to narrow down the one or more virtual machines you wish to query, and then pipe the results to Get-RubrikMount.

### PARAMETERS

**-id <String>** Rubrik’s id of the mount

**-VMID <String>** Filters live mounts by VM ID

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

EXAMPLE 1

```
PS C:>Get-RubrikMount
```

This will return details on all mounted virtual machines.

EXAMPLE 2

```
PS C:>Get-RubrikMount -id '11111111-2222-3333-4444-555555555555'
```

This will return details on mount id “11111111-2222-3333-4444-555555555555”.

EXAMPLE 3

```
PS C:>Get-RubrikMount -VMID (Get-RubrikVM -VM 'Server1').id
```

This will return details for any mounts found using the id value from a virtual machine named “Server1” as a base reference.

EXAMPLE 4

```
PS C:>Get-RubrikMount -VMID 'VirtualMachine:::aaaaaaa-bbbb-cccc-dddd-eeeeeeeeeee-vm-12345'
```

This will return details for any mounts found using the virtual machine id of ‘VirtualMachine:::aaaaaaa-bbbb-cccc-dddd-eeeeeeeeeee-vm-12345’ as a base reference.

**REMARKS** To see the examples, type: “get-help Get-RubrikMount -examples”. For more information, type: “get-help Get-RubrikMount -detailed”. For technical information, type: “get-help Get-RubrikMount -full”. For online help, type: “get-help Get-RubrikMount -online”

## Get-RubrikReport

**NAME** Get-RubrikReport

**SYNOPSIS** Retrieves details on one or more reports created in Rubrik Envision

**SYNTAX** Get-RubrikReport [[-Name] <String>] [[-Type] <String>] [[-id] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikReport cmdlet is used to pull information on any number of Rubrik Envision reports

### PARAMETERS

**-Name <String>** Filter the returned reports based off their name.

**-Type <String>** Filter the returned reports based off the reports type. Options are Canned and Custom.

**-id <String>** The ID of the report.

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikReport
```

This will return details on all reports

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikReport -Name 'SLA' -Type Custom
```

This will return details on all custom reports that contain the string “SLA”

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikReport -id '11111111-2222-3333-4444-555555555555'
```

This will return details on the report id “11111111-2222-3333-4444-555555555555”

**REMARKS** To see the examples, type: “get-help Get-RubrikReport -examples”. For more information, type: “get-help Get-RubrikReport -detailed”. For technical information, type: “get-help Get-RubrikReport -full”. For online help, type: “get-help Get-RubrikReport -online”

## Get-RubrikReportData

**NAME** Get-RubrikReportData

**SYNOPSIS** Retrieve table data for a specific Envision report

**SYNTAX** Get-RubrikReportData [-id] <String> [[-Name] <String>] [[-TaskType] <String>] [[-TaskStatus] <String>] [[-ObjectType] <String>] [[-ComplianceStatus] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikReportData cmdlet is used to pull table data from a specific Envision report

**PARAMETERS**

- id <String> The ID of the report
  - Name <String> Search table data by object name
  - TaskType <String> Filter table data on task type
  - TaskStatus <String> Filter table data on task status
  - ObjectType <String> Filter table data on object type
  - ComplianceStatus <String> Filter table data on compliance status
  - Server <String> Rubrik server IP or FQDN
  - api <String> API version
- <CommonParameters> This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikReport -Name 'SLA Compliance Summary' | Get-RubrikReportData
```

This will return table data from the “SLA Compliance Summary” report

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikReport -Name 'SLA Compliance Summary' | Get-RubrikReportData -ComplianceStatus 'NonCompliance'
```

This will return table data from the “SLA Compliance Summary” report when the compliance status is “Non-Compliance”

**REMARKS** To see the examples, type: “get-help Get-RubrikReportData -examples”. For more information, type: “get-help Get-RubrikReportData -detailed”. For technical information, type: “get-help Get-RubrikReportData -full”. For online help, type: “get-help Get-RubrikReportData -online”

## Get-RubrikRequest

**NAME** Get-RubrikRequest

**SYNOPSIS** Connects to Rubrik and retrieves details on an async request

**SYNTAX** Get-RubrikRequest [-id] <String> [-Type] <String> [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikRequest cmdlet will pull details on a request that was submitted to the distributed task framework. This is helpful for tracking the state (success, failure, running, etc.) of a request.

**PARAMETERS**

- id <String> ID of an asynchronous request
- Type <String> The type of request
- Server <String> Rubrik server IP or FQDN
- api <String> API version



**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikRequest -id 'MOUNT_SNAPSHOT_123456789:::0' -Type 'vmware/vm'
```

Will return details about an async VMware VM request named “MOUNT\_SNAPSHOT\_123456789:::0”

**REMARKS** To see the examples, type: “get-help Get-RubrikRequest -examples”. For more information, type: “get-help Get-RubrikRequest -detailed”. For technical information, type: “get-help Get-RubrikRequest -full”. For online help, type: “get-help Get-RubrikRequest -online”

## Get-RubrikSLA

**NAME** Get-RubrikSLA

**SYNOPSIS** Connects to Rubrik and retrieves details on SLA Domain(s)

**SYNTAX** Get-RubrikSLA [[-Name] <String>] [[-PrimaryClusterID] <String>] [[-id] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikSLA cmdlet will query the Rubrik API for details on all available SLA Domains. Information on each domain will be reported to the console.

### PARAMETERS

**-Name <String>** Name of the SLA Domain

**-PrimaryClusterID <String>** Filter the summary information based on the primarycluster\_id of the primary Rubrik cluster. Use **\_local** as the primary\_cluster\_id of the Rubrik cluster that is hosting the current REST API session.

**-id <String>** SLA Domain id

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikSLA
```

Will return all known SLA Domains

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikSLA -Name 'Gold'
```

Will return details on the SLA Domain named Gold

**REMARKS** To see the examples, type: “get-help Get-RubrikSLA -examples”. For more information, type: “get-help Get-RubrikSLA -detailed”. For technical information, type: “get-help Get-RubrikSLA -full”. For online help, type: “get-help Get-RubrikSLA -online”

## Get-RubrikSnapshot

**NAME** Get-RubrikSnapshot

**SYNOPSIS** Retrieves all of the snapshots (backups) for any given object

**SYNTAX** Get-RubrikSnapshot [-id] <String> [[-CloudState] <Int32>] [-OnDemandSnapshot] [[-Date] <DateTime>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikSnapshot cmdlet is used to query the Rubrik cluster for all known snapshots (backups) for any protected object The correct API call will be made based on the object id submitted Multiple objects can be piped into this function so long as they contain the id required for lookup

### PARAMETERS

**-id** <String> Rubrik id of the protected object

**-CloudState** <Int32> Filter results based on where in the cloud the snapshot lives

**-OnDemandSnapshot** [<SwitchParameter>] Filter results to show only snapshots that were created on demand

**-Date** <DateTime> Date of the snapshot

**-Server** <String> Rubrik server IP or FQDN

**-api** <String> API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

#### EXAMPLE 1

```
PS C:>Get-RubrikSnapshot -id 'VirtualMachine:::aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeee-vm-12345'
```

This will return all snapshot (backup) data for the virtual machine id of "VirtualMachine:::aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeee-vm-12345"

#### EXAMPLE 2

```
PS C:>Get-Rubrikvm 'Server1' | Get-RubrikSnapshot -Date '03/21/2017'
```

This will return the closest matching snapshot to March 21st, 2017 for any virtual machine named "Server1"

#### EXAMPLE 3

```
PS C:>Get-Rubrikvm 'Server1' | Get-RubrikSnapshot -Date (Get-Date)
```

This will return the closest matching snapshot to the current date and time for any virtual machine named "Server1"

#### EXAMPLE 4

```
PS C:>Get-RubrikDatabase 'DB1' | Get-RubrikSnapshot -OnDemandSnapshot
```

This will return the details on any on-demand (user initiated) snapshot to for any database named "DB1"

**REMARKS** To see the examples, type: "get-help Get-RubrikSnapshot -examples". For more information, type: "get-help Get-RubrikSnapshot -detailed". For technical information, type: "get-help Get-RubrikSnapshot -full". For online help, type: "get-help Get-RubrikSnapshot -online"

## Get-RubrikSoftwareVersion

**NAME** Get-RubrikSoftwareVersion

**SYNOPSIS** Connects to Rubrik and retrieves the current software version

**SYNTAX** Get-RubrikSoftwareVersion [-Server] <String> [[-id] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikVersion cmdlet will retrieve the version of code that is actively running on the system. This does not require authentication.

### PARAMETERS

**-Server <String>** Rubrik server IP or FQDN

**-id <String>** ID of the Rubrik cluster or me for self

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikSoftwareVersion -Server 192.168.1.100
```

This will return the running software version on the Rubrik cluster reachable at the address 192.168.1.100

**REMARKS** To see the examples, type: “get-help Get-RubrikSoftwareVersion -examples”. For more information, type: “get-help Get-RubrikSoftwareVersion -detailed”. For technical information, type: “get-help Get-RubrikSoftwareVersion -full”. For online help, type: “get-help Get-RubrikSoftwareVersion -online”

## Get-RubrikUnmanagedObject

**NAME** Get-RubrikUnmanagedObject

**SYNOPSIS** Retrieves details on one or more unmanaged objects known to a Rubrik cluster

**SYNTAX** Get-RubrikUnmanagedObject [[-Name] <String>] [[-Status] <String>] [[-Type] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikUnmanagedObject cmdlet is used to pull details on any unmanaged objects that has been stored in the cluster In most cases, this will be on-demand snapshots that are associated with an object (virtual machine, fileset, database, etc.)

### PARAMETERS

**-Name <String>** Search object by object name.

**-Status <String>** Filter by the type of the object. If not specified, will return all objects. Valid attributes are Protected, Relic and Unprotected

**-Type <String>** The type of the unmanaged object. This may be VirtualMachine, MssqlDatabase, LinuxFileset, or WindowsFileset.

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikUnmanagedObject -Type 'WindowsFileset'
```

This will return details on any filesets applied to Windows Servers that have unmanaged snapshots associated

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikUnmanagedObject -Status 'Unprotected' -Name 'Server1'
```

This will return details on any objects named “Server1” that are currently unprotected and have unmanaged snapshots associated

**REMARKS** To see the examples, type: “get-help Get-RubrikUnmanagedObject -examples”. For more information, type: “get-help Get-RubrikUnmanagedObject -detailed”. For technical information, type: “get-help Get-RubrikUnmanagedObject -full”. For online help, type: “get-help Get-RubrikUnmanagedObject -online”

## Get-RubrikVersion

**NAME** Get-RubrikVersion

**SYNOPSIS** Connects to Rubrik and retrieves the current version

**SYNTAX** Get-RubrikVersion [[-id] <String>] [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikVersion cmdlet will retrieve the version of code that is actively running on the system.

**PARAMETERS**

**-id <String>** ID of the Rubrik cluster or me for self

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikVersion
```

This will return the running version on the Rubrik cluster

**REMARKS** To see the examples, type: “get-help Get-RubrikVersion -examples”. For more information, type: “get-help Get-RubrikVersion -detailed”. For technical information, type: “get-help Get-RubrikVersion -full”. For online help, type: “get-help Get-RubrikVersion -online”

## Get-RubrikVM

**NAME** Get-RubrikVM

**SYNOPSIS** Retrieves details on one or more virtual machines known to a Rubrik cluster

**SYNTAX** Get-RubrikVM [[-Name] <String>] [-Relic] [-SLA <String>] [-SLAAssignment <String>] [-PrimaryClusterID <String>] [-id <String>] [-SLAID <String>] [-Server <String>] [-api <String>] [<CommonParameters>]

**DESCRIPTION** The Get-RubrikVM cmdlet is used to pull a detailed data set from a Rubrik cluster on any number of virtual machines

## PARAMETERS

**-Name <String>** Name of the virtual machine

**-Relic [<SwitchParameter>]** Filter results to include only relic (removed) virtual machines

**-SLA <String>** SLA Domain policy assigned to the virtual machine

**-SLAAssignment <String>** Filter by SLA Domain assignment type

**-PrimaryClusterID <String>** Filter the summary information based on the primarycluster\_id of the primary Rubrik cluster. Use **\_local** as the primary\_cluster\_id of the Rubrik cluster that is hosting the current REST API session.

**-id <String>** Virtual machine id

**-SLAID <String>** SLA id value

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikVM -Name 'Server1'
```

This will return details on all virtual machines named “Server1”.

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikVM -Name 'Server1' -SLA Gold
```

This will return details on all virtual machines named “Server1” that are protected by the Gold SLA Domain.

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikVM -Relic
```

This will return all removed virtual machines that were formerly protected by Rubrik.

**REMARKS** To see the examples, type: “get-help Get-RubrikVM -examples”. For more information, type: “get-help Get-RubrikVM -detailed”. For technical information, type: “get-help Get-RubrikVM -full”. For online help, type: “get-help Get-RubrikVM -online”



This page contains details on **Move** commands.

## Move-RubrikMountVMDK

**NAME** Move-RubrikMountVMDK

**SYNOPSIS** Moves the VMDKs from a Live Mount to another VM

**SYNTAX** Move-RubrikMountVMDK [-SourceVM] <String> [-TargetVM] <String> [[-Date] <String>] [[-ExcludeDisk] <Array>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Move-RubrikMountVMDK [-Cleanup <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Move-RubrikMountVMDK cmdlet is used to attach VMDKs from a Live Mount to another VM, typically for restore or testing purposes.

### PARAMETERS

**-SourceVM <String>** Source virtual machine to use as a Live Mount based on a previous backup

**-TargetVM <String>** Target virtual machine to attach the Live Mount disk(s)

**-Date <String>** Backup date to use for the Live Mount Will use the current date and time if no value is specified

**-ExcludeDisk <Array>** An array of disks to exclude from presenting to the target virtual machine By default, all disks will be presented

**-Cleanup <String>** The path to a cleanup file to remove the live mount and presented disks The cleanup file is created each time the command is run and stored in the \$HOME path as a text file with a random number value The file contains the TargetVM name, MountID value, and a list of all presented disks

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

-WhatIf [<SwitchParameter>]

-Confirm [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Move-RubrikMountVMDK -SourceVM 'SourceVM' -TargetVM 'TargetVM'
```

This will create a Live Mount using the latest snapshot of the VM named “SourceVM” The Live Mount’s VMDKs would then be presented to the VM named “TargetVM”

————— EXAMPLE 2 —————

```
PS C:>Move-RubrikMountVMDK -SourceVM 'SourceVM' -TargetVM 'TargetVM' -Date '01/30/2016 08:00'
```

This will create a Live Mount using the January 30th 08:00AM snapshot of the VM named “SourceVM” The Live Mount’s VMDKs would then be presented to the VM named “TargetVM” Note: The Date parameter will start at the time specified (in this case, 08:00am) and work backwards in time until it finds a snapshot. Precise timing is not required.

————— EXAMPLE 3 —————

```
PS C:>Move-RubrikMountVMDK -SourceVM 'SourceVM' -TargetVM 'TargetVM' -ExcludeDisk @(0,1)
```

This will create a Live Mount using the latest snapshot of the VM named “SourceVM” Disk 0 and 1 (the first and second disks) would be excluded from presentation to the VM named “TargetVM” Note: that for the “ExcludeDisk” array, the format is @(##,##,...) where each # represents a disk starting with 0. Example: To exclude the first and third disks, the value would be @(0,2). Example: To exclude just the first disk, use @(0).

————— EXAMPLE 4 —————

```
PS C:>Move-RubrikMountVMDK -Cleanup 'C:\Users\Person1\Documents\SourceVM_to_TargetVM-1234567890.txt'
```

This will remove the disk(s) and live mount, effectively reversing the initial request This file is created each time the command is run and stored in the \$HOME path as a text file The file contains the TargetVM name, MountID value, and a list of all presented disks

**REMARKS** To see the examples, type: “get-help Move-RubrikMountVMDK -examples”. For more information, type: “get-help Move-RubrikMountVMDK -detailed”. For technical information, type: “get-help Move-RubrikMountVMDK -full”. For online help, type: “get-help Move-RubrikMountVMDK -online”



This page contains details on **New** commands.

## New-RubrikDatabaseMount

**NAME** New-RubrikDatabaseMount

**SYNOPSIS** Create a new Live Mount from a protected database

**SYNTAX** New-RubrikDatabaseMount -id <String> [-TargetInstanceId <String>] [-MountedDatabaseName <String>] [-TimestampMs <Int64>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

New-RubrikDatabaseMount -id <String> [-TargetInstanceId <String>] [-MountedDatabaseName <String>] [-RecoveryDateTime <DateTime>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The New-RubrikDatabaseMount cmdlet is used to create a Live Mount (clone) of a protected database and run it in an existing database environment.

### PARAMETERS

- id <String>** Rubrik id of the database
- TargetInstanceId <String>** ID of Instance to use for the mount
- MountedDatabaseName <String>** Name of the mounted database
- TimestampMs <Int64>** Recovery Point desired in the form of Epoch with Milliseconds
- RecoveryDateTime <DateTime>** Recovery Point desired in the form of DateTime value
- Server <String>** Rubrik server IP or FQDN
- api <String>** API version

-WhatIf [<SwitchParameter>]

-Confirm [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>New-RubrikDatabaseMount -id $db.id -targetInstanceId $db.instanceId -mountedDatabaseName 'BAR-LM' -recoveryDateTime (Get-date (Get-RubrikDatabase -id $db.id).latestRecoveryPoint)
```

Creates a new database mount named BAR on the same instance as the source database, using the most recent recovery time for the database.

```
$db=Get-RubrikDatabase -HostName FOO -Instance MSSQLSERVER -Database BAR
```

**REMARKS** To see the examples, type: “get-help New-RubrikDatabaseMount -examples”. For more information, type: “get-help New-RubrikDatabaseMount -detailed”. For technical information, type: “get-help New-RubrikDatabaseMount -full”. For online help, type: “get-help New-RubrikDatabaseMount -online”

## New-RubrikHost

**NAME** New-RubrikHost

**SYNOPSIS** Registers a host with a Rubrik cluster.

**SYNTAX** New-RubrikHost [-Name] <String> [[-HasAgent] <Boolean>] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The New-RubrikHost cmdlet is used to register a host with the Rubrik cluster. This could be a host leveraging the Rubrik Backup Service or directly as with the case of NAS shares.

### PARAMETERS

**-Name <String>** The IPv4 address of the host or the resolvable hostname of the host

**-HasAgent <Boolean>** Set to \$false to register a host that will be accessed through network shares

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

-WhatIf [<SwitchParameter>]

-Confirm [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>New-RubrikHost -Name 'Server1.example.com'
```

This will register a host that resolves to the name “Server1.example.com”

————— EXAMPLE 2 —————

```
PS C:>New-RubrikHost -Name 'NAS.example.com' -HasAgent $false
```

This will register a host that resolves to the name “NAS.example.com” without using the Rubrik Backup Service. In this case, the example host is a NAS share.

**REMARKS** To see the examples, type: “get-help New-RubrikHost -examples”. For more information, type: “get-help New-RubrikHost -detailed”. For technical information, type: “get-help New-RubrikHost -full”. For online help, type: “get-help New-RubrikHost -online”

## New-RubrikMount

**NAME** New-RubrikMount

**SYNOPSIS** Create a new Live Mount from a protected VM

**SYNTAX** New-RubrikMount [-id] <String> [[-HostID] <String>] [[-MountName] <String>] [[-DatastoreName] <String>] [[-DisableNetwork] <Boolean>] [-RemoveNetworkDevices] [-PowerOn] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The New-RubrikMount cmdlet is used to create a Live Mount (clone) of a protected VM and run it in an existing vSphere environment.

### PARAMETERS

**-id** <String> Rubrik id of the snapshot

**-HostID** <String> ID of host for the mount to use

**-MountName** <String> Name of the mounted VM

**-DatastoreName** <String> Name of the data store to use/create on the host

**-DisableNetwork** <Boolean> Whether the network should be disabled on mount. This should be set true to avoid ip conflict in case of static IPs.

**-RemoveNetworkDevices** [<SwitchParameter>] Whether the network devices should be removed on mount.

**-PowerOn** [<SwitchParameter>] Whether the VM should be powered on after mount.

**-Server** <String> Rubrik server IP or FQDN

**-api** <String> API version

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>New-RubrikMount -id '11111111-2222-3333-4444-555555555555'
```

This will create a new mount based on snapshot id “11111111-2222-3333-4444-555555555555” The original virtual machine’s name will be used along with a date and index number suffix The virtual machine will NOT be powered on upon completion of the mount operation

————— EXAMPLE 2 —————

```
PS C:>New-RubrikMount -id '11111111-2222-3333-4444-555555555555' -MountName 'Mount1' -PowerOn
-RemoveNetworkDevices
```

This will create a new mount based on snapshot id “11111111-2222-3333-4444-555555555555” and name the mounted virtual machine “Mount1” The virtual machine will be powered on upon completion of the mount operation but without any virtual network adapters

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikVM 'Server1' | Get-RubrikSnapshot -Date '03/01/2017 01:00' | New-RubrikMount -MountName 'Mount1' -DisableNetwork
```

This will create a new mount based on the closet snapshot found on March 1st, 2017 @ 01:00 AM and name the mounted virtual machine “Mount1” The virtual machine will NOT be powered on upon completion of the mount operation

**REMARKS** To see the examples, type: “get-help New-RubrikMount -examples”. For more information, type: “get-help New-RubrikMount -detailed”. For technical information, type: “get-help New-RubrikMount -full”. For online help, type: “get-help New-RubrikMount -online”

## New-RubrikReport

**NAME** New-RubrikReport

**SYNOPSIS** Create a new report by specifying one of the report templates

**SYNTAX** New-RubrikReport [-Name] <String> [-ReportTemplate] <String> [[-Server] <String>] [[-api] <String>] [<CommonParameters>]

**DESCRIPTION** The New-RubrikReport cmdlet is used to create a new Envision report by specifying one of the canned report templates

### PARAMETERS

**-Name <String>** The name of the report

**-ReportTemplate <String>** The template this report is based on

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>New-RubrikReport -Name 'Report1' -ReportTemplate 'ProtectionTasksDetails'
```

This will create a new report named “Report1” by using the “ProtectionTasksDetails” report template

**REMARKS** To see the examples, type: “get-help New-RubrikReport -examples”. For more information, type: “get-help New-RubrikReport -detailed”. For technical information, type: “get-help New-RubrikReport -full”. For online help, type: “get-help New-RubrikReport -online”

## New-RubrikSLA

**NAME** New-RubrikSLA

**SYNOPSIS** Creates a new Rubrik SLA Domain

**SYNTAX** New-RubrikSLA [-Name] <String> [[-HourlyFrequency] <Int32>] [[-HourlyRetention] <Int32>] [[-DailyFrequency] <Int32>] [[-DailyRetention] <Int32>] [[-MonthlyFrequency] <Int32>] [[-MonthlyRetention] <Int32>] [[-YearlyFrequency] <Int32>] [[-YearlyRetention] <Int32>] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The New-RubrikSLA cmdlet will build a new SLA Domain to provide policy-driven control over protected objects within the Rubrik fabric.

**PARAMETERS**

- Name <String> SLA Domain Name
- HourlyFrequency <Int32> Hourly frequency to take backups
- HourlyRetention <Int32> Number of hours to retain the hourly backups
- DailyFrequency <Int32> Daily frequency to take backups
- DailyRetention <Int32> Number of days to retain the daily backups
- MonthlyFrequency <Int32> Monthly frequency to take backups
- MonthlyRetention <Int32> Number of months to retain the monthly backups
- YearlyFrequency <Int32> Yearly frequency to take backups
- YearlyRetention <Int32> Number of years to retain the yearly backups
- Server <String> Rubrik server IP or FQDN
- api <String> API version
- WhatIf [<SwitchParameter>]
- Confirm [<SwitchParameter>]

<CommonParameters> This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>New-RubrikSLA -SLA 'Test1' -HourlyFrequency 4 -HourlyRetention 24
```

This will create an SLA Domain named “Test1” that will take a backup every 4 hours and keep those hourly backups for 24 hours.

————— EXAMPLE 2 —————

```
PS C:>New-RubrikSLA -SLA 'Test1' -HourlyFrequency 4 -HourlyRetention 24 -DailyFrequency 1 -
DailyRetention 30
```

This will create an SLA Domain named “Test1” that will take a backup every 4 hours and keep those hourly backups for 24 hours while also keeping one backup per day for 30 days.

**REMARKS** To see the examples, type: “get-help New-RubrikSLA -examples”. For more information, type: “get-help New-RubrikSLA -detailed”. For technical information, type: “get-help New-RubrikSLA -full”. For online help, type: “get-help New-RubrikSLA -online”

## New-RubrikSnapshot

**NAME** New-RubrikSnapshot

**SYNOPSIS** Takes an on-demand Rubrik snapshot of a protected object

**SYNTAX** New-RubrikSnapshot -id <String> [-SLA <String>] [-ForceFull] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

```
New-RubrikSnapshot -id <String> [-DoNotProtect] [-ForceFull] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

```
New-RubrikSnapshot -id <String> [-Inherit] [-ForceFull] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**DESCRIPTION** The New-RubrikSnapshot cmdlet will trigger an on-demand snapshot for a specific object (virtual machine, database, fileset, etc.)

**PARAMETERS**

- id <String> Rubrik’s id of the object
- SLA <String> The SLA Domain in Rubrik
- DoNotProtect [<SwitchParameter>] Removes the SLA Domain assignment
- Inherit [<SwitchParameter>] Inherits the SLA Domain assignment from a parent object
- ForceFull [<SwitchParameter>] Whether to force a full snapshot or an incremental. Only valid with MSSQL Databases.
- SLAID <String> SLA id value
- Server <String> Rubrik server IP or FQDN
- api <String> API version
- WhatIf [<SwitchParameter>]
- Confirm [<SwitchParameter>]
- <CommonParameters> This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikVM ‘Server1’ | New-RubrikSnapshot
```

This will trigger an on-demand backup for any virtual machine named “Server1”

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikFileset ‘C_Drive’ | New-RubrikSnapshot -SLA ‘Gold’
```

This will trigger an on-demand backup for any fileset named “C\_Drive” using the “Gold” SLA Domain

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikDatabase ‘DB1’ | New-RubrikSnapshot -ForceFull
```

This will trigger an on-demand backup for any database named “DB1” and force the backup to be a full rather than an incremental.

**REMARKS** To see the examples, type: “get-help New-RubrikSnapshot -examples”. For more information, type: “get-help New-RubrikSnapshot -detailed”. For technical information, type: “get-help New-RubrikSnapshot -full”. For online help, type: “get-help New-RubrikSnapshot -online”

---

## Protect Commands

---

This page contains details on **Protect** commands.

### Protect-RubrikDatabase

**NAME** Protect-RubrikDatabase

**SYNOPSIS** Connects to Rubrik and assigns an SLA to a database

**SYNTAX** Protect-RubrikDatabase -id <String> [-SLA <String>] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Protect-RubrikDatabase -id <String> [-DoNotProtect] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Protect-RubrikDatabase -id <String> [-Inherit] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Protect-RubrikDatabase cmdlet will update a database's SLA Domain assignment within the Rubrik cluster. The SLA Domain contains all policy-driven values needed to protect workloads. Note that this function requires the Database ID value, not the name of the database, since database names are not unique across hosts. It is suggested that you first use Get-RubrikDatabase to narrow down the one or more database / instance / hosts to protect, and then pipe the results to Protect-RubrikDatabase. You will be asked to confirm each database you wish to protect, or you can use -Confirm:\$False to skip confirmation checks.

#### PARAMETERS

**-id <String>** Database ID

**-SLA <String>** The SLA Domain in Rubrik

**-DoNotProtect [<SwitchParameter>]** Removes the SLA Domain assignment

**-Inherit [<SwitchParameter>]** Inherits the SLA Domain assignment from a parent object

**-SLAID <String>** SLA id value

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf [<SwitchParameter>]**

**-Confirm [<SwitchParameter>]**

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikDatabase "DB1" | Protect-RubrikDatabase -SLA 'Gold'
```

This will assign the Gold SLA Domain to any database named "DB1"

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikDatabase "DB1" -Instance "MSSQLSERVER" | Protect-RubrikDatabase -SLA 'Gold' -Confirm:$False
```

This will assign the Gold SLA Domain to any database named "DB1" residing on an instance named "MSSQLSERVER" without asking for confirmation

**REMARKS** To see the examples, type: "get-help Protect-RubrikDatabase -examples". For more information, type: "get-help Protect-RubrikDatabase -detailed". For technical information, type: "get-help Protect-RubrikDatabase -full". For online help, type: "get-help Protect-RubrikDatabase -online"

## Protect-RubrikFileset

**NAME** Protect-RubrikFileset

**SYNOPSIS** Connects to Rubrik and assigns an SLA to a fileset

**SYNTAX** Protect-RubrikFileset -id <String> [-SLA <String>] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Protect-RubrikFileset -id <String> [-DoNotProtect] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Protect-RubrikFileset cmdlet will update a fileset's SLA Domain assignment within the Rubrik cluster. The SLA Domain contains all policy-driven values needed to protect data. Note that this function requires the fileset ID value, not the name of the fileset, since fileset names are not unique across clusters. It is suggested that you first use Get-RubrikFileset to narrow down the one or more filesets to protect, and then pipe the results to Protect-RubrikFileset. You will be asked to confirm each fileset you wish to protect, or you can use -Confirm:\$False to skip confirmation checks.

### PARAMETERS

**-id <String>** Fileset ID

**-SLA <String>** The SLA Domain in Rubrik

**-DoNotProtect [<SwitchParameter>]** Removes the SLA Domain assignment

**-SLAID <String>** SLA id value

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version



-WhatIf [**<SwitchParameter>**]

-Confirm [**<SwitchParameter>**]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikFileset 'C_Drive' | Protect-RubrikFileset -SLA 'Gold'
```

This will assign the Gold SLA Domain to any fileset named “C\_Drive”

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikFileset 'C_Drive' -HostName 'Server1' | Protect-RubrikFileset -SLA 'Gold' -Confirm:$False
```

This will assign the Gold SLA Domain to the fileset named “C\_Drive” residing on the host named “Server1” without asking for confirmation

**REMARKS** To see the examples, type: “get-help Protect-RubrikFileset -examples”. For more information, type: “get-help Protect-RubrikFileset -detailed”. For technical information, type: “get-help Protect-RubrikFileset -full”. For online help, type: “get-help Protect-RubrikFileset -online”

## Protect-RubrikTag

**NAME** Protect-RubrikTag

**SYNOPSIS** Connects to Rubrik and assigns an SLA Domain based on a vSphere category and tag value

**SYNTAX** Protect-RubrikTag -Tag <String> -Category <String> [-SLA <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [**<CommonParameters>**]

```
Protect-RubrikTag -Tag <String> -Category <String> [-DoNotProtect] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

```
Protect-RubrikTag -Tag <String> -Category <String> [-Inherit] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**DESCRIPTION** The Protect-RubrikTag cmdlet will update a virtual machine’s SLA Domain assignment within the Rubrik cluster. The SLA Domain contains all policy-driven values needed to protect workloads. Make sure you have PowerCLI installed and connect to the required vCenter Server.

### PARAMETERS

**-Tag <String>** vSphere Tag

**-Category <String>** vSphere Tag Category

**-SLA <String>** The SLA Domain in Rubrik

**-DoNotProtect [**<SwitchParameter>**]** Removes the SLA Domain assignment

**-Inherit [**<SwitchParameter>**]** Inherits the SLA Domain assignment from a parent object

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

-WhatIf [**<SwitchParameter>**]

-Confirm [**<SwitchParameter>**]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Protect-RubrikTag -Tag 'Gold' -Category 'Rubrik' -SLA 'Gold'
```

This will assign the Gold SLA Domain to any VM tagged with Gold in the Rubrik category

————— EXAMPLE 2 —————

```
PS C:>Protect-RubrikTag -Tag 'Gold' -Category 'Rubrik' -SLA 'Titanium'
```

This will assign the Titanium SLA Domain to any VM tagged with Gold in the Rubrik category

————— EXAMPLE 3 —————

```
PS C:>Protect-RubrikTag -Tag 'Gold' -Category 'Rubrik' -DoNotProtect
```

This will remove protection from any VM tagged with Gold in the Rubrik category

————— EXAMPLE 4 —————

```
PS C:>Protect-RubrikTag -Tag 'Gold' -Category 'Rubrik' -Inherit
```

This will flag any VM tagged with Gold in the Rubrik category to inherit the SLA Domain of its parent object

**REMARKS** To see the examples, type: “get-help Protect-RubrikTag -examples”. For more information, type: “get-help Protect-RubrikTag -detailed”. For technical information, type: “get-help Protect-RubrikTag -full”. For online help, type: “get-help Protect-RubrikTag -online”

## Protect-RubrikVM

**NAME** Protect-RubrikVM

**SYNOPSIS** Connects to Rubrik and assigns an SLA to a virtual machine

**SYNTAX** Protect-RubrikVM -id <String> [-SLA <String>] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

```
Protect-RubrikVM -id <String> [-DoNotProtect] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

```
Protect-RubrikVM -id <String> [-Inherit] [-SLAID <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

**DESCRIPTION** The Protect-RubrikVM cmdlet will update a virtual machine’s SLA Domain assignment within the Rubrik cluster. The SLA Domain contains all policy-driven values needed to protect workloads. Note that this function requires the virtual machine ID value, not the name of the virtual machine, since virtual machine names are not unique across clusters. It is suggested that you first use Get-RubrikVM to narrow down the one or more virtual machine to protect, and then pipe the results to Protect-RubrikVM. You will be asked to confirm each virtual machine you wish to protect, or you can use -Confirm:\$False to skip confirmation checks.

### PARAMETERS

**-id <String>** Virtual machine ID

**-SLA <String>** The SLA Domain in Rubrik

**-DoNotProtect [<SwitchParameter>]** Removes the SLA Domain assignment

**-Inherit [<SwitchParameter>]** Inherits the SLA Domain assignment from a parent object

**-SLAID** <String> SLA id value  
**-Server** <String> Rubrik server IP or FQDN  
**-api** <String> API version  
**-WhatIf** [<SwitchParameter>]  
**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikVM "VM1" | Protect-RubrikVM -SLA 'Gold'
```

This will assign the Gold SLA Domain to any virtual machine named "VM1"

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikVM "VM1" -SLA Silver | Protect-RubrikVM -SLA 'Gold' -Confirm:$False
```

This will assign the Gold SLA Domain to any virtual machine named "VM1" that is currently assigned to the Silver SLA Domain without asking for confirmation

**REMARKS** To see the examples, type: "get-help Protect-RubrikVM -examples". For more information, type: "get-help Protect-RubrikVM -detailed". For technical information, type: "get-help Protect-RubrikVM -full". For online help, type: "get-help Protect-RubrikVM -online"



---

## Remove Commands

---

This page contains details on **Remove** commands.

### Remove-RubrikDatabaseMount

**NAME** Remove-RubrikDatabaseMount

**SYNOPSIS** Connects to Rubrik and removes one or more database live mounts

**SYNTAX** Remove-RubrikDatabaseMount [-id] <String> [-Force] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Remove-RubrikDatabaseMount cmdlet is used to request the deletion of one or more instant database mounts

#### PARAMETERS

**-id <String>** The Rubrik ID value of the mount

**-Force [<SwitchParameter>]** Force unmount to deal with situations where host has been moved.

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf [<SwitchParameter>]**

**-Confirm [<SwitchParameter>]**

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

---

#### EXAMPLE 1

---

```
PS C:>Remove-RubrikDatabaseMount -id '11111111-2222-3333-4444-555555555555'
```

This will remove mount id "11111111-2222-3333-4444-555555555555".

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikDatabaseMount | Remove-RubrikDatabaseMount
```

This will remove all mounted databases.

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikDatabaseMount -source_database_name 'BAR' | Remove-RubrikDatabaseMount
```

This will remove any mounts found using the datase name as a base reference.

**REMARKS** To see the examples, type: “get-help Remove-RubrikDatabaseMount -examples”. For more information, type: “get-help Remove-RubrikDatabaseMount -detailed”. For technical information, type: “get-help Remove-RubrikDatabaseMount -full”. For online help, type: “get-help Remove-RubrikDatabaseMount -online”

## Remove-RubrikFileset

**NAME** Remove-RubrikFileset

**SYNOPSIS** Delete a fileset by specifying the fileset ID

**SYNTAX** Remove-RubrikFileset [-id] <String> [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Remove-RubrikFileset cmdlet is used to remove a fileset registered with the Rubrik cluster.

### PARAMETERS

**-id <String>** The Rubrik ID value of the fileset

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikFileset -Name 'C_Drive' | Remove-RubrikHost
```

This will remove any fileset that matches the name “C\_Drive”

————— EXAMPLE 2 —————

```
PS C:>Remove-RubrikFileset -id 'Fileset:::111111-2222-3333-4444-555555555555'
```

This will specifically remove the fileset id matching “Fileset:::111111-2222-3333-4444-555555555555”

**REMARKS** To see the examples, type: “get-help Remove-RubrikFileset -examples”. For more information, type: “get-help Remove-RubrikFileset -detailed”. For technical information, type: “get-help Remove-RubrikFileset -full”. For online help, type: “get-help Remove-RubrikFileset -online”

## Remove-RubrikHost

**NAME** Remove-RubrikHost

**SYNOPSIS** Delete host by specifying the host ID.

**SYNTAX** Remove-RubrikHost [-id] <String> [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Remove-RubrikHost cmdlet is used to remove a host registered with the Rubrik cluster.

### PARAMETERS

**-id <String>** The Rubrik ID value of the host

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikHost -Name 'Server1.example.com' | Remove-RubrikHost
```

This will remove a host that matches the name "Server1.example.com"

————— EXAMPLE 2 —————

```
PS C:>Remove-RubrikHost -id 'Host:::111111-2222-3333-4444-555555555555'
```

This will specifically remove the host id matching "Host:::111111-2222-3333-4444-555555555555"

**REMARKS** To see the examples, type: "get-help Remove-RubrikHost -examples". For more information, type: "get-help Remove-RubrikHost -detailed". For technical information, type: "get-help Remove-RubrikHost -full". For online help, type: "get-help Remove-RubrikHost -online"

## Remove-RubrikMount

**NAME** Remove-RubrikMount

**SYNOPSIS** Connects to Rubrik and removes one or more live mounts

**SYNTAX** Remove-RubrikMount [-id] <String> [-Force] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Remove-RubrikMount cmdlet is used to request the deletion of one or more instant mounts

### PARAMETERS

**-id <String>** The Rubrik ID value of the mount

**-Force** [<SwitchParameter>] Force unmount to deal with situations where host has been moved.

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

-WhatIf [<SwitchParameter>]

-Confirm [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Remove-RubrikMount -id '11111111-2222-3333-4444-555555555555'
```

This will remove mount id “11111111-2222-3333-4444-555555555555”.

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikMount | Remove-RubrikMount
```

This will remove all mounted virtual machines.

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikMount -VMID (Get-RubrikVM -VM 'Server1').id | Remove-RubrikMount
```

This will remove any mounts found using the virtual machine named “Server1” as a base reference.

**REMARKS** To see the examples, type: “get-help Remove-RubrikMount -examples”. For more information, type: “get-help Remove-RubrikMount -detailed”. For technical information, type: “get-help Remove-RubrikMount -full”. For online help, type: “get-help Remove-RubrikMount -online”

## Remove-RubrikReport

**NAME** Remove-RubrikReport

**SYNOPSIS** Removes one or more reports created in Rubrik Envision

**SYNTAX** Remove-RubrikReport [-id] <String> [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Remove-RubrikReport cmdlet is used to delete any number of Rubrik Envision reports

### PARAMETERS

**-id <String>** The Rubrik ID value of the report

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

-WhatIf [<SwitchParameter>]

-Confirm [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikReport | Remove-RubrikReport -Confirm:$true
```

This will delete all reports and force confirmation for each delete operation

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikReport -Name 'SLA' -Type Custom | Remove-RubrikReport
```



This will delete all custom reports that contain the string “SLA”

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikReport -id '11111111-2222-3333-4444-555555555555' | Remove-RubrikReport -
Confirm:$false
```

This will delete the report id “11111111-2222-3333-4444-555555555555” without confirmation

**REMARKS** To see the examples, type: “get-help Remove-RubrikReport -examples”. For more information, type: “get-help Remove-RubrikReport -detailed”. For technical information, type: “get-help Remove-RubrikReport -full”. For online help, type: “get-help Remove-RubrikReport -online”

## Remove-RubrikSLA

**NAME** Remove-RubrikSLA

**SYNOPSIS** Connects to Rubrik and removes SLA Domains

**SYNTAX** Remove-RubrikSLA [-id] <String> [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Remove-RubrikSLA cmdlet will request that the Rubrik API delete an SLA Domain. The SLA Domain must have zero protected objects (VMs, filesets, databases, etc.) in order to be successful.

### PARAMETERS

**-id** <String> SLA Domain id

**-Server** <String> Rubrik server IP or FQDN

**-api** <String> API version

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikSLA -SLA 'Gold' | Remove-RubrikSLA
```

This will attempt to remove the Gold SLA Domain from Rubrik if there are no objects being protected by the policy

**REMARKS** To see the examples, type: “get-help Remove-RubrikSLA -examples”. For more information, type: “get-help Remove-RubrikSLA -detailed”. For technical information, type: “get-help Remove-RubrikSLA -full”. For online help, type: “get-help Remove-RubrikSLA -online”

## Remove-RubrikUnmanagedObject

**NAME** Remove-RubrikUnmanagedObject

**SYNOPSIS** Removes one or more unmanaged objects known to a Rubrik cluster

**SYNTAX** Remove-RubrikUnmanagedObject [-id] <String> [-Type] <String> [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Remove-RubrikUnmanagedObject cmdlet is used to remove unmanaged objects that have been stored in the cluster. In most cases, this will be on-demand snapshots that are associated with an object (virtual machine, fileset, database, etc.)

**PARAMETERS**

**-id <String>** The id of the unmanaged object.

**-Type <String>** The type of the unmanaged object. This may be VirtualMachine, MssqlDatabase, LinuxFileset, or WindowsFileset.

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf [<SwitchParameter>]**

**-Confirm [<SwitchParameter>]**

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikUnmanagedObject | Remove-RubrikUnmanagedObject
```

This will remove all unmanaged objects from the cluster

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikUnmanagedObject -Type 'WindowsFileset' | Remove-RubrikUnmanagedObject -Confirm:$false
```

This will remove any unmanaged objects related to filesets applied to Windows Servers and suppress confirmation for each activity

————— EXAMPLE 3 —————

```
PS C:>Get-RubrikUnmanagedObject -Status 'Unprotected' -Name 'Server1' | Remove-RubrikUnmanagedObject
```

This will remove any unmanaged objects associated with any workload named "Server1" that is currently unprotected

**REMARKS** To see the examples, type: "get-help Remove-RubrikUnmanagedObject -examples". For more information, type: "get-help Remove-RubrikUnmanagedObject -detailed". For technical information, type: "get-help Remove-RubrikUnmanagedObject -full". For online help, type: "get-help Remove-RubrikUnmanagedObject -online"

---

## Restore Commands

---

This page contains details on **Restore** commands.

### Restore-RubrikDatabase

**NAME** Restore-RubrikDatabase

**SYNOPSIS** Connects to Rubrik and restores a MSSQL database

**SYNTAX** Restore-RubrikDatabase -Id <String> [-MaxDataStreams <Int32>] [-TimestampMs <Int64>] [-FinishRecovery] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

Restore-RubrikDatabase -Id <String> [-MaxDataStreams <Int32>] [-RecoveryDateTime <DateTime>] [-FinishRecovery] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Restore-RubrikDatabase command will request a database restore from a Rubrik Cluster to a MSSQL instance. This is an inplace restore, meaning it will overwrite the existing asset.

#### PARAMETERS

- Id <String> Rubrik identifier of database to be exported
- MaxDataStreams <Int32> Number of parallel streams to copy data
- TimestampMs <Int64> Recovery Point desired in the form of Epoch with Milliseconds
- RecoveryDateTime <DateTime> Recovery Point desired in the form of DateTime value
- FinishRecovery [<SwitchParameter>] If FinishRecover is true, fully recover the database
- Server <String> Rubrik server IP or FQDN
- api <String> API version
- WhatIf [<SwitchParameter>]
- Confirm [<SwitchParameter>]

<**CommonParameters**> This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Restore-RubrikDatabase -id MssqlDatabase:::c5ecf3ef-248d-4bb2-8fe1-4d3c820a0e38 -targetInstanceId MssqlInstance:::0085b247-e718-4177-869f-e3ae1f7bb503 -FinishRecovery -maxDataStreams 4 -timestampMs 1492661627000
```

Restore database to declared epoch ms timestamp.

————— EXAMPLE 2 —————

```
PS C:>Restore-RubrikDatabase -id $db.id -recoveryDateTime (Get-Date (Get-RubrikDatabase $db).latestRecoveryPoint) -maxDataStreams 1 -FinishRecovery
```

Restore the \$db (where \$db is the output of a Get-RubrikDatabase call) to the most recent recovery point for that database.

**REMARKS** To see the examples, type: “get-help Restore-RubrikDatabase -examples”. For more information, type: “get-help Restore-RubrikDatabase -detailed”. For technical information, type: “get-help Restore-RubrikDatabase -full”. For online help, type: “get-help Restore-RubrikDatabase -online”

This page contains details on **Set** commands.

## Set-RubrikBlackout

**NAME** Set-RubrikBlackout

**SYNOPSIS** Connects to Rubrik and sets blackout (stops/starts all snaps)

**SYNTAX** Set-RubrikBlackout [-Set] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Set-RubrikBlackout cmdlet will accept a flag of true/false to set cluster blackout

### PARAMETERS

**-Set** [<SwitchParameter>] Rubrik blackout value

**-Server** <String> Rubrik server IP or FQDN

**-api** <String> API version

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

---

### EXAMPLE 1

```
PS C:>Set-RubrikBlackout -Set [true/false]
```

**REMARKS** To see the examples, type: “get-help Set-RubrikBlackout -examples”. For more information, type: “get-help Set-RubrikBlackout -detailed”. For technical information, type: “get-help Set-RubrikBlackout -full”. For online help, type: “get-help Set-RubrikBlackout -online”

## Set-RubrikDatabase

**NAME** Set-RubrikDatabase

**SYNOPSIS** Sets Rubrik database properties

**SYNTAX** Set-RubrikDatabase [-id <String>] [-LogBackupFrequencyInSeconds <Int32>] [-LogRetentionHours <Int32>] [-CopyOnly] [-MaxDataStreams <Int32>] [-SLAID <String>] [-SLA <String>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Set-RubrikDatabase cmdlet is used to update certain settings for a Rubrik database.

### PARAMETERS

**-id <String>** Rubrik's database id value

**-LogBackupFrequencyInSeconds <Int32>** Number of seconds between log backups if db is in FULL or BULK\_LOGGED

**-LogRetentionHours <Int32>** Number of hours backups will be retained in Rubrik

**-CopyOnly [<SwitchParameter>]** Boolean declaration for copy only backups on the database.

**-MaxDataStreams <Int32>** Number of max data streams Rubrik will use to back up the database

**-SLAID <String>** SLA Domain ID for the database

**-SLA <String>** The SLA Domain name in Rubrik

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf [<SwitchParameter>]**

**-Confirm [<SwitchParameter>]**

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

#### EXAMPLE 1

```
PS C:>Set-RubrikDatabase -id MssqlDatabase:::c5ecf3ef-248d-4bb2-8fe1-4d3c820a0e38 -LogBackupFrequencyInSeconds 900
```

Set the target database's log backup interval to 15 minutes (900 seconds)

#### EXAMPLE 2

```
PS C:>Get-RubrikDatabase -HostName Foo -Instance MSSQLSERVER | Set-RubrikDatabase -SLA 'Silver' -CopyOnly
```

Set all databases on host FOO to use SLA Silver and be copy only.

**REMARKS** To see the examples, type: “get-help Set-RubrikDatabase -examples”. For more information, type: “get-help Set-RubrikDatabase -detailed”. For technical information, type: “get-help Set-RubrikDatabase -full”. For online help, type: “get-help Set-RubrikDatabase -online”

## Set-RubrikMount

**NAME** Set-RubrikMount

**SYNOPSIS** Powers on/off a live mounted virtual machine within a connected Rubrik vCenter.

**SYNTAX** Set-RubrikMount [-id] <String> [-PowerOn <Boolean>] [-Server <String>] [-api <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Set-RubrikMount cmdlet is used to send a power on request to mounted virtual machine visible to a Rubrik cluster.

#### PARAMETERS

**-id <String>** Mount id

**-PowerOn <Boolean>** Configuration for the change power status request

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

-WhatIf [<SwitchParameter>]

-Confirm [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

#### EXAMPLE 1

```
PS C:>Get-RubrikMount -id '11111111-2222-3333-4444-555555555555' | Set-RubrikMount -PowerOn:$true
```

This will send a power on request to “Server1”

#### EXAMPLE 2

```
PS C:>Get-RubrikMount -VMID (Get-RubrikVM -VM 'Server1').id | Set-RubrikMount -PowerOn:$false
```

This will send a power off request to “Server1”

**REMARKS** To see the examples, type: “get-help Set-RubrikMount -examples”. For more information, type: “get-help Set-RubrikMount -detailed”. For technical information, type: “get-help Set-RubrikMount -full”. For online help, type: “get-help Set-RubrikMount -online”

## Set-RubrikVM

**NAME** Set-RubrikVM

**SYNOPSIS** Applies settings on one or more virtual machines known to a Rubrik cluster

**SYNTAX** Set-RubrikVM [-id] <String> [[-SnapConsistency] <String>] [[-MaxNestedSnapshots] <Int32>] [[-PauseBackups] <Boolean>] [[-UseArrayIntegration] <Boolean>] [[-Server] <String>] [[-api] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

**DESCRIPTION** The Set-RubrikVM cmdlet is used to apply updated settings from a Rubrik cluster on any number of virtual machines

#### PARAMETERS

**-id <String>** Virtual machine ID

**-SnapConsistency <String>** Consistency level mandated for this VM

**-MaxNestedSnapshots <Int32>** The number of existing virtual machine snapshots allowed by Rubrik. Choices range from 0 - 4 snapshots.

**-PauseBackups <Boolean>** Whether to pause or resume backups/archival for this VM.

**-UseArrayIntegration <Boolean>** User setting to dictate whether to use storage array snapshots for ingest. This setting only makes sense for VMs where array based ingest is possible.

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**-WhatIf** [<SwitchParameter>]

**-Confirm** [<SwitchParameter>]

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Get-RubrikVM 'Server1' | Set-RubrikVM -PauseBackups
```

This will pause backups on any virtual machine named “Server1”

————— EXAMPLE 2 —————

```
PS C:>Get-RubrikVM -SLA Platinum | Set-RubrikVM -SnapConsistency 'CRASH_CONSISTENT' -MaxNestedSnapshots 2 -UseArrayIntegration
```

This will find all virtual machines in the Platinum SLA Domain and set their snapshot consistency to crash consistent (no application quiescence) while also limiting the number of active hypervisor snapshots to 2 and enable storage array (SAN) snapshots for ingest

**REMARKS** To see the examples, type: “get-help Set-RubrikVM -examples”. For more information, type: “get-help Set-RubrikVM -detailed”. For technical information, type: “get-help Set-RubrikVM -full”. For online help, type: “get-help Set-RubrikVM -online”



This page contains details on **Sync** commands.

## Sync-RubrikAnnotation

**NAME** Sync-RubrikAnnotation

**SYNOPSIS** Applies Rubrik SLA Domain information to VM Annotations using the Custom Attributes feature in vCenter

**SYNTAX** Sync-RubrikAnnotation [[-SLA] <String>] [[-SLAAnnotationName] <String>] [[-BackupAnnotationName] <String>] [-Server <String>] [-api <String>] [<CommonParameters>]

**DESCRIPTION** The Sync-RubrikAnnotation cmdlet will comb through all VMs currently being protected by Rubrik. It will then create Custom Attribute buckets for SLA Domain Name(s) and Snapshot counts and assign details for each VM found in vCenter using Annotations. The attribute names can be specified using this function's parameters or left as the defaults. See the examples for more information. Keep in mind that this only displays in the VMware vSphere Thick (C#) client, which is deprecated moving forward.

### PARAMETERS

- SLA <String>** Optional filter for a single SLA Domain Name By default, all SLA Domain Names will be collected when this parameter is not used
- SLAAnnotationName <String>** Attribute name in vCenter for the Rubrik SLA Domain Name By default, will use "Rubrik\_SLA"
- BackupAnnotationName <String>** Attribute name in vCenter for quantity of snapshots By default, will use "Rubrik\_Backups"
- Server <String>** Rubrik server IP or FQDN
- api <String>** API version
- <CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Sync-RubrikAnnotation
```

This will find all VMs being protected with any Rubrik SLA Domain Name and update their SLA and snapshot count annotations using the defaults of “Rubrik\_SLA” and “Rubrik\_Backups” respectively.

————— EXAMPLE 2 —————

```
PS C:>Sync-RubrikAnnotation -SLA Silver
```

This will find all VMs being protected with a Rubrik SLA Domain Name of “Silver” and update their SLA and snapshot count annotations using the defaults of “Rubrik\_SLA” and “Rubrik\_Backups” respectively.

————— EXAMPLE 3 —————

```
PS C:>Sync-RubrikAnnotation -SLAAnnotationName 'Backup-Policy' -BackupAnnotationName 'Backup-Snapshots'
```

This will find all VMs being protected with any Rubrik SLA Domain Name and update their SLA and snapshot count annotations using the custom values of “Backup-Policy” and “Backup-Snapshots” respectively.

**REMARKS** To see the examples, type: “get-help Sync-RubrikAnnotation -examples”. For more information, type: “get-help Sync-RubrikAnnotation -detailed”. For technical information, type: “get-help Sync-RubrikAnnotation -full”. For online help, type: “get-help Sync-RubrikAnnotation -online”

## Sync-RubrikTag

**NAME** Sync-RubrikTag

**SYNOPSIS** Connects to Rubrik and creates a vSphere tag for each SLA Domain

**SYNTAX** Sync-RubrikTag [-Category] <String> [-Server <String>] [-api <String>] [<CommonParameters>]

**DESCRIPTION** The Sync-RubrikTag cmdlet will query Rubrik for all of the existing SLA Domains, and then create a tag for each one

**PARAMETERS**

**-Category <String>** The vSphere Category name for the Rubrik SLA Tags

**-Server <String>** Rubrik server IP or FQDN

**-api <String>** API version

**<CommonParameters>** This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<http://go.microsoft.com/fwlink/?LinkID=113216>).

————— EXAMPLE 1 —————

```
PS C:>Sync-RubrikTag -vCenter 'vcenter1.demo' -Category 'Rubrik'
```

This will validate or create a vSphere Category named Rubrik along with a Tag for each SLA Domain found in Rubrik

**REMARKS** To see the examples, type: “get-help Sync-RubrikTag -examples”. For more information, type: “get-help Sync-RubrikTag -detailed”. For technical information, type: “get-help Sync-RubrikTag -full”. For online help, type: “get-help Sync-RubrikTag -online”

This page contains workflows to audit Rubrik activities.

### Failed Tasks

The activity log contains detailed information on every task. In the event of a failure, a number of options are available to the administrator - alerts, traps, logs, and so forth. If you wish to pull this data for programmatic use, the following workflow can be used:

```
New-RubrikReport -ReportType daily -StatusType Failed
```

The resulting data will contain a key named `failureDescription` that can be parsed for more information. A snippet sample value is shown below:

```
failureDescription : Rubrik backup service at 'object' returned error: Remote call_
↳exception: object:12801 error: SSL_read: Resource temporarily unavailable
status             : Failed
jobType            : Backup
```

Storing this information into a variable, CSV file, or other object can be helpful for pushing failed task data to 3rd party systems.

**CSV Export:** `New-RubrikReport -ReportType daily -StatusType Failed | Export-Csv -Encoding UTF8 -Path $Home\Documents\Report.CSV -NoTypeInfoation`

**Store to Variable:** `$Report = New-RubrikReport -ReportType daily -StatusType Failed`



## CHAPTER 21

---

### Backup Validation Workflows

---