ArmorPowerShell Documentation

Release stable

Jun 08, 2018

User Documentation

This is a community project that provides a powerful command-line interface for managing and monitoring your Armor Complete (secure public cloud) and Armor Anywhere (security as a service) environments & accounts via a PowerShell module with cmdlets that interact with the published RESTful APIs.

Every code push is built on **Windows** via AppVeyor, as well as on **macOS** and **Ubuntu Linux** via Travis CI, and tested using the Pester test & mock framework.

Code coverage scores and reports showing how much of the project is covered by automated tests are tracked by Coveralls.

Every successful build is published on the PowerShell Gallery.

The source code is *available on GitHub*.

CHAPTER 1

Requirements

At a minimum, you will need the following:

1. An Armor Complete or Armor Anywhere account.
2. An Armor Management Portal user account that has been granted membership to one or more roles configured with sufficient permissions to interact with the environment.
3. Either:

1. Windows PowerShell version 5.0 or
2. Higher or PowerShell Core version 6.0 or higher.

CHAPTER 2

Install

This repository contains a folder named Armor. 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 terminal window.

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\`

2.1 Option 1: PowerShell Gallery Installation (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 Armor -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.

2.2 Option 2: PowerShell Gallery Download

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 `Save-Module -Name Armor -Path <path>` 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.

5. Copy the contents of the Armor module folder onto your workstation into the desired PowerShell Module path.

2.3 Option 3: Manual Installation

1. Download the master branch to your workstation.
2. Copy the contents of the *Armor* 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*.

2.4 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 Armor Module are installed:

```
Get-Module -ListAvailable -Name Armor
```

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 -Name Armor
```

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

```
Import-Module -Name Armor
```

If you wish to load a specific version, use:

```
Import-Module -Name Armor -RequiredVersion #.#.#.#
```

Where "#.#.#.#" represents the version number.

CHAPTER 3

Update

3.1 Option 1: PowerShell Gallery Update (Recommended)

If you installed the module via the PowerShell Gallery, please implement the following when you want to update to a newer version:

1. Open a Powershell console with the *Run as Administrator* option.
2. Run `Update-Module -Name Armor`.

3.2 Option 2: PowerShell Gallery Download

If you deployed a saved module via the PowerShell Gallery, please implement the following when you want to update to a newer version:

1. Open a Powershell console with the *Run as Administrator* option.
2. Run `Save-Module -Name Armor -Path <path>` 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.
3. Copy the contents of the Armor module folder onto your workstation into the desired PowerShell Module path.

3.3 Option 3: Manual Installation

If you deployed the module via download from GitHub, please implement the following when you want to update to a newer version:

1. Download the master branch to your workstation.
2. Copy the contents of the *Armor* folder onto your workstation into the desired PowerShell Module path.

CHAPTER 4

Uninstall

4.1 Option 1: Uninstall-Module (Recommended)

If you installed the module via the PowerShell Gallery, please implement the following when you want to uninstall the module:

1. Open a Powershell console with the *Run as Administrator* option.
2. Run `Uninstall-Module -Name Armor`.

4.2 Option 2: Delete the Armor module directory

1. Delete the Armor directory in your PowerShell Module path.

CHAPTER 5

Getting Started

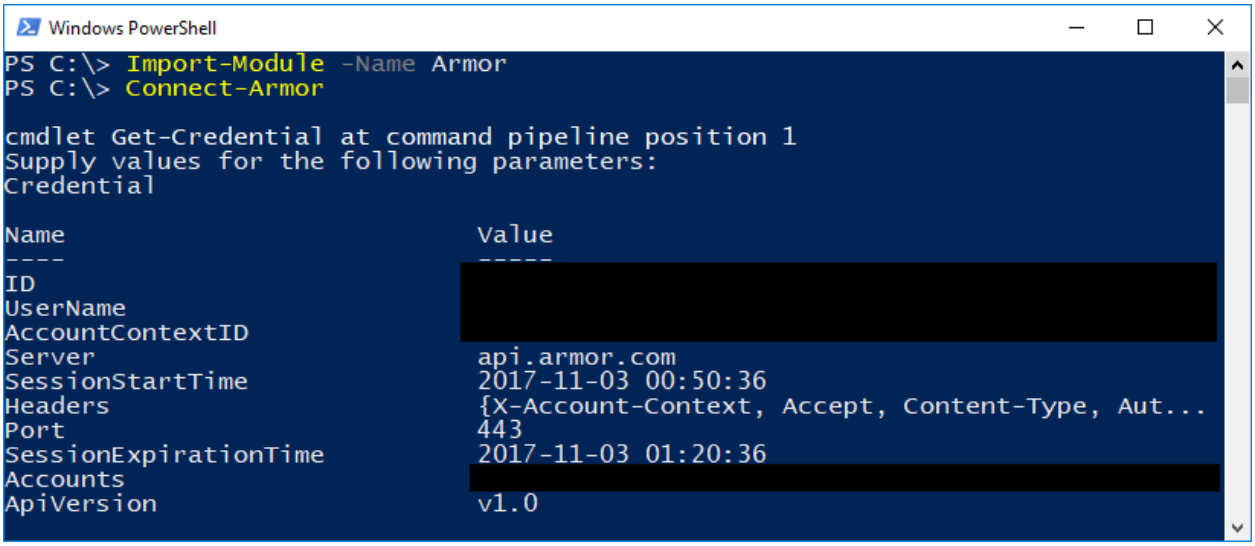
Now that you have the Armor module installed on your workstation, there's a few beginner commands that you can explore to feel comfortable with the available functions.

5.1 Connecting to Armor

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

- Open a PowerShell terminal window
- Type `Connect-Armor` and press enter.

A prompt will appear asking you for credentials. Enter your Armor Management Portal (AMP) username and password. Once entered, you will receive a multi-factor authentication request. Once accepted, you will see details about the newly created connection.



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

5.2 Commands and Help

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

- Adheres to the Microsoft requirements for PowerShell functions.
- Use of "Armor" in the name should avoid collisions with other commands.

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

```
Get-Help -Name Connect-Armor
```

This will display a description about the command.

For details and examples, use the `-Detailed` parameter.

```
Get-Help Connect-Armor -Detailed
```

For all available information, use the `-Full` parameter.

```
Get-Help Connect-Armor -Full
```

5.3 Gathering Data

Let's get information about the Armor account. 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 Armor cluster. Enter the command below and press enter:

```
Get-ArmorAccount
```

The result is fairly simple: the command will output the Armor accounts that this user account has access to. How about something a bit more complex? Try getting all of the virtual machines (VMs) in the Armor account. Here's the command:

```
Get-ArmorVM
```

Lots of stuff should be scrolling across the screen. You're seeing information on every Armor VM at a very detailed level. If you want to see just one Armor VM, 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-ArmorVM -ID 12345
```

The `-ID` portion is a parameter and 12345 is a value for the parameter. This effectively asks the function to limit results to the specified VM: 12345. Easy, right?

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

5.4 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 Armor cluster. To validate this, use the following command:

```
Get-ArmorVM -Name "Name"
```

Make sure to replace `"Name"` with the actual name of the virtual machine. If you received data back from Armor, you can be sure that this virtual machine exists in the account and can be modified if the user has sufficient permissions.

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 rename this virtual machine. To do this, use the following command:

```
Rename-ArmorVM -ID 12345 -Name "NewName"
```

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 "Renames the specified virtual machine in your account" on␣
↪target "20931"".
[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:

```
Rename-ArmorVM -ID 12345 -Name "NewName" -Confirm:$false
```

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

CHAPTER 6

Project Architecture

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

- **Armor: The parent folder containing the module**
  - **Formats**: Output decoration definition files
  - **Etc**: Configuration files
  - **Lib**: Class files
  - **Private**: Private functions that are used internally by the module
  - **Public**: Published functions that are available to the user when the PowerShell module is loaded
  - **Armor.psd1**: Module manifest
  - **Armor.psm1**: Script module file
- **build**: Continuous integration initialization and build scripts
- **deploy**: Continuous deployment scripts to publish to the PowerShell Gallery and GitHub
- **docs**: Module documentation
- **templates**: Templates for creating your own functions
- **tests: Pester unit tests used to validate the public functions**
  - **config**: Continuous integration environment and module configuration tests
  - **etc**: Configuration files
  - **lib**: Class tests
  - **private**: Private function tests
  - **public**: Public function tests
- **workflows**: Sample workflows for more complex automation tasks

CHAPTER 7

Support

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

A future release may include formal support.

To report a bug, request an enhancement, or provide feedback about this project, please open an issue.

CHAPTER 8

Licensing

CHAPTER 9

Frequently Asked Questions

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

CHAPTER 10

Connect Commands

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

10.1 Connect-Armor

NAME Connect-Armor

SYNOPSIS Connects to the Armor API and establishes a session.

SYNTAX Connect-Armor [[-Credential] <PSCredential>] [[-AccountID] <UInt16>] [[-Server] <String>] [[-Port] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Connects to the Armor RESTful API and supplies credentials to the method. The Armor API then returns a unique, temporary authorization code, which is then converted into a token to represent the user's credentials for subsequent calls. Last, the account context is set. If an account ID is not specified, one is automatically selected from the list of authorized account IDs. Returns the session details which are stored in the variable: $Global:ArmorSession.

PARAMETERS

-Credential <PSCredential> Your Armor API username and password. If not supplied as a parameter, you will be prompted for your credentials.

-AccountID <UInt16> Specifies the Armor account ID to use for all subsequent requests. The permitted range is 1-65535.

-Server <String> Specifies the Armor API server IP address or FQDN.

-Port <UInt16> Specifies the Armor API server listening TCP port. The permitted range is: 1-65535.

-ApiVersion <String> Specifies the API version for this request. The specified value is also set as the default API version for the session as a parameter of the session variable: '$Global:ArmorSession.ApiVersion'.

The API version can be specified when any other public cmdlets are called or the value of '$Global:ArmorSession.ApiVersion' can be updated afterward to set a different default API version for the session.

<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-Armor

Prompts for the username and password, and then attempts to log into the Armor API.

———————— EXAMPLE 2 ————————

PS C:>Connect-Armor -Credential $pscredential

Attempts to log into the Armor API with the credentials stored in the $pscredential object.

———————— EXAMPLE 3 ————————

PS C:>Connect-Armor -Credential $pscredential -AccountID 12345

Attempts to log into the Armor API with the credentials stored in the $pscredential object, and sets the account context to '12345'.

———————— EXAMPLE 4 ————————

PS C:>Connect-Armor -Credential $pscredential -ApiVersion 'v1.0'

Attempts to log into the Armor API with the credentials stored in the $pscredential object and sets the specified API version as the default for the session, which is stored in $Global:ArmorSession.ApiVersion.

———————— EXAMPLE 5 ————————

PS C:>Connect-Armor -Credential $pscredential -Server 'localhost' -Port 8443

Attempts to log into a local test/dev Armor API instance listening on port 8443/tcp with the credentials stored in the $pscredential object.

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

CHAPTER 11

Disconnect Commands

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

11.1 Disconnect-Armor

NAME Disconnect-Armor

SYNOPSIS Disconnects from Armor and destroys the session information.

SYNTAX Disconnect-Armor [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION Disconnects from the Armor API and destroys the $Global:ArmorSession session variable.

PARAMETERS -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-Armor

Disconnects from the Armor API and destroys the $Global:ArmorSession session variable.

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

CHAPTER 12

Get Commands

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

12.1 Get-ArmorAccount

NAME Get-ArmorAccount

SYNOPSIS Retrieves Armor account details.

SYNTAX Get-ArmorAccount [[-ID] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorAccount [[-Name] <String>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Retrieves a list of Armor account memberships for the currently authenticated user. Returns a set of accounts that correspond to the filter criteria provided by the cmdlet parameters.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor account.

-Name <String> Specifies the name of the Armor account.

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorAccount

Gets all Armor accounts assigned to the logged in user account.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorAccount -Name *Child*

Gets all Armor accounts assigned to the logged in user account with a name containing the word 'Child'.

———————— EXAMPLE 3 ————————

PS C:>1, 'Example Child Account' | Get-ArmorAccount

Gets the Armor accounts assigned to the logged in user account with ID=1 and Name='Example Child Account' via pipeline values.

———————— EXAMPLE 4 ————————

PS C:>[PSCustomObject] @{ 'ID' = 1 } | Get-ArmorAccount

Gets the Armor account assigned to the logged in user account with ID=1 via property name in the pipeline.

———————— EXAMPLE 5 ————————

PS C:>[PSCustomObject] @{ 'Name' = 'My Secure Account' } | Get-ArmorAccount

Gets the Armor account assigned to the logged in user account with Name='My Secure Account' via property name in the pipeline.

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

12.2 Get-ArmorAccountAddress

NAME Get-ArmorAccountAddress

SYNOPSIS Retrieves the mailing address on file for Armor accounts.

SYNTAX Get-ArmorAccountAddress [[-ID] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION This cmdlet retrieves the mailing address on file for Armor accounts that your user account has access to.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor account with the desired address details.

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorAccountAddress

Retrieves the mailing address of the Armor account currently in context.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorAccountAddress -ID 1

Retrieves the mailing address of the Armor account with ID 1.

———————— EXAMPLE 3 ————————

PS C:>1, 2 | Get-ArmorAccountAddress

Retrieves the mailing address of the Armor accounts with ID=1 and ID=2 via pipeline values.

———————— EXAMPLE 4 ————————

PS C:>[PSCustomObject] @{ 'ID' = 1 } | Get-ArmorAccountAddress

Retrieves the mailing address of the Armor account with ID=1 and ID=2 via property names in the pipeline.

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

12.3 Get-ArmorAccountContext

NAME Get-ArmorAccountContext

SYNOPSIS Retrieves the Armor Anywhere or Armor Complete account currently in context.

SYNTAX Get-ArmorAccountContext [<CommonParameters>]

DESCRIPTION If your user account has access to more than one Armor Anywhere and/or Armor Complete accounts, this cmdlet allows you to get the current context, which all future requests will reference.

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:>Get-ArmorAccountContext

Retrieves the Armor account currently in context.

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

12.4 Get-ArmorCompleteDatacenter

NAME Get-ArmorCompleteDatacenter

SYNOPSIS Retrieves Armor Complete datacenter details.

SYNTAX Get-ArmorCompleteDatacenter [[-ID] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorCompleteDatacenter [-Name <String>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorCompleteDatacenter [[-Location] <String>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Retrieves details about the Armor Complete datacenters, regions, and compute zones. Returns a set of datacenters that correspond to the filter criteria provided by the cmdlet parameters.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete datacenter.

-Name <String> Specifies the name of the Armor Complete region.

-Location <String> Specifies the name of the Armor Complete datacenter.

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorCompleteDatacenter

Retrieves the details for all Armor Complete datacenters.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorCompleteDatacenter -ID 2

Retrieves the details for the Armor Complete datacenter with ID=2.

———————— EXAMPLE 3 ————————

PS C:>1, 'PHX01' | Get-ArmorCompleteDatacenter

Retrieves the details for the Armor Complete datacenter with ID=1 and Location='PHX01' via pipeline values.

———————— EXAMPLE 4 ————————

PS C:>[PSCustomObject] @{ 'Location' = 'EU West' } | Get-ArmorCompleteDatacenter

Retrieves the details for the Armor Complete datacenter with Name='EU West' via property name in the pipeline.

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

12.5 Get-ArmorCompleteWorkload

NAME Get-ArmorCompleteWorkload

SYNOPSIS This cmdlet retrieves Armor Complete workloads.

SYNTAX Get-ArmorCompleteWorkload [[-ID] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorCompleteWorkload [[-Name] <String>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Workloads and tiers are logical grouping tools for helping you organize your virtual machines and corresponding resources in your Armor Complete software-defined datacenters.

Workloads contain tiers, and tiers contain virtual machines.

Workloads are intended to help you describe the business function of a group of servers, such as 'My Secure Website', which could be useful for chargeback or showback to your customers, as well as helping your staff and the Armor Support teams understand the architecture of your environment.

Tiers are intended to describe the application tiers within each workload. A typical three tiered application workload is comprised of presentation, business logic, and persistence tiers. Common labels for each are: web, application, and database respectively, but you can group your VMs however you choose.

Returns a set of workloads that correspond to the filter criteria provided by the cmdlet parameters.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete workload.

-Name <String> Specifies the name of the Armor Complete workload.

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorCompleteWorkload

Retrieves the details for all workloads in the Armor Complete account that currently has context.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorCompleteWorkload -ID 1

Retrieves the details for the workload with ID=1.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorCompleteWorkload -Name 'LAMP stack'

Retrieves the details for the workload with Name='LAMP stack'.

———————— EXAMPLE 4 ————————

PS C:>2, 'WISP stack' | Get-ArmorCompleteWorkload -ApiVersion 'v1.0'

Retrieves the API version 1.0 details for the workloads with ID=2 and Name='WISP stack' via pipeline values.

———————— EXAMPLE 5 ————————

PS C:>[PSCustomObject] @{ 'Name' = 'Secure stack' } | Get-ArmorCompleteWorkload

Retrieves the details for the workload with Name='Secure stack' via property name in the pipeline.

———————— EXAMPLE 6 ————————

PS C:>[PSCustomObject] @{ 'ID' = 1 } | Get-ArmorCompleteWorkload

Retrieves the details for the workload with ID=1 via property name in the pipeline.

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

12.6 Get-ArmorCompleteWorkloadTier

NAME Get-ArmorCompleteWorkloadTier

SYNOPSIS This cmdlet retrieves the tiers in an Armor Complete workload.

SYNTAX Get-ArmorCompleteWorkloadTier [-WorkloadID] <UInt16> [[-ID] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorCompleteWorkloadTier [-WorkloadID] <UInt16> [[-Name] <String>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Workloads and tiers are logical grouping tools for helping you organize your virtual machines and corresponding resources in your Armor Complete software-defined datacenters.

Workloads contain tiers, and tiers contain virtual machines.

Workloads are intended to help you describe the business function of a group of servers, such as 'My Secure Website', which could be useful for chargeback or showback to your customers, as well as helping your staff and the Armor Support teams understand the architecture of your environment.

Tiers are intended to describe the application tiers within each workload. A typical three tiered application workload is comprised of presentation, business logic, and persistence tiers. Common labels for each are: web, application, and database respectively, but you can group your VMs however you choose.

Returns a set of tiers in a workload that correspond to the filter criteria provided by the cmdlet parameters.

PARAMETERS

-WorkloadID <UInt16> Specifies the ID of the workload that contains the tier(s).

-ID <UInt16> Specifies the ID of the workload tier.

-Name <String> Specifies the names of the workload tiers.

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorCompleteWorkloadTier -WorkloadID 1

Retrieves the details for all workload tiers in the workload with WorkloadID=1 in the Armor Complete account that currently has context.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorCompleteWorkloadTier -WorkloadID 1 -ID 1

Retrieves the details for the workload tier with ID=1 in the workload with WorkloadID=1.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorCompleteWorkloadTier -WorkloadID 1 -Name 'Database'

Retrieves the details for the workload tier with Name='Database' in the workload with WorkloadID=1.

———————— EXAMPLE 4 ————————

PS C:>2, 3 | Get-ArmorCompleteWorkloadTier -ApiVersion 'v1.0'

Retrieves the API version 1.0 details for all of the workload tiers in workloads with WorkloadID=2 and WorkloadID=3 via pipeline values.

———————— EXAMPLE 5 ————————

PS C:>[PSCustomObject] @{ 'WorkloadID' = 1; 'ID' = 1 } | Get-ArmorCompleteWorkloadTier

Retrieves the details for the workload tier with ID=1 in the workload with WorkloadID=1 via property names in the pipeline.

———————— EXAMPLE 6 ————————

PS C:>[PSCustomObject] @{ 'WorkloadID' = 1; 'Name' = 'Presentation' } | Get-ArmorCompleteWorkloadTier

Retrieves the details for the workload tier with Name='Presentation' in the workload with WorkloadID=1 via property names in the pipeline.

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

12.7 Get-ArmorIdentity

NAME Get-ArmorIdentity

SYNOPSIS Retrieves identity details about your Armor user account.

SYNTAX Get-ArmorIdentity [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Retrieves details about your Armor user account that you used to establish the session, including account membership and permissions.

This also updates the identity information in the session variable: $Global:ArmorSession.

PARAMETERS

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorIdentity

Retrieves the identity details about your Armor user account.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorIdentity -ApiVersion 1.0

Retrieves the Armor API version 1.0 identity details about your Armor user account.

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

12.8 Get-ArmorUser

NAME Get-ArmorUser

SYNOPSIS Retrieves details about the user accounts in your account.

SYNTAX Get-ArmorUser [[-ID] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorUser [[-UserName] <String>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorUser [-FirstName <String>] [-LastName <String>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Retrieves details about the user accounts in the Armor Anywhere or Armor Complete account in context. Returns a set of user accounts that correspond to the filter criteria provided by the cmdlet parameters.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor user account.

-UserName <String> Specifies the username of the Armor user account.

-FirstName <String> Specifies the first name of the Armor user account.

-LastName <String> Specifies the last name of the Armor user account.

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorUser

Retrieves the details for all user accounts in the Armor account that currently has context.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorUser -ID 1

Retrieves the details for all user accounts in the Armor account that currently has context.

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

12.9 Get-ArmorVM

NAME Get-ArmorVM

SYNOPSIS Retrieves virtual machine details.

SYNTAX Get-ArmorVM [[-ID] <UInt16>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorVM [[-CoreInstanceID] <Guid>] [[-ApiVersion] <String>] [<CommonParameters>]

Get-ArmorVM [[-Name] <String>] [[-ApiVersion] <String>] [<CommonParameters>]

DESCRIPTION Retrieves details about the virtual machines in the Armor Anywhere or Armor Complete account in context. Returns a set of virtual machines that correspond to the filter criteria provided by the cmdlet parameters.

PARAMETERS

-ID <UInt16> Specifies the IDs of the virtual machines that you want to retrieve.

-CoreInstanceID <Guid> Specifies the Armor Anywhere Core Agent instance IDs of the virtual machines that you want to retrieve.

-Name <String> Specifies the names of the virtual machines that you want to retrieve.

-ApiVersion <String> Specifies the API version for this request.

<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-ArmorVM

Retrieves the details for all VMs in the Armor account that currently has context.

———————— EXAMPLE 2 ————————

PS C:>Get-ArmorVM -ID 1

Retrieves the details for the VM with ID=1.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorVM -Name 'web1'

Retrieves the details for the VM with Name='web1'.

———————— EXAMPLE 4 ————————

PS C:>Get-ArmorVM -Name db*

Retrieves all VMs in the Armor account that currently has context that have a name that starts with 'db'.

———————— EXAMPLE 5 ————————

PS C:>1 | Get-ArmorVM

Retrieves the details for the VM with ID=1 via pipeline value.

———————— EXAMPLE 6 ————————

PS C:>'*secure*' | Get-ArmorVM

Retrieves all VMs containing the word 'secure' in the name via pipeline value.

———————— EXAMPLE 7 ————————

PS C:>[PSCustomObject] @{ 'ID' = 1 } | Get-ArmorVM

Retrieves the details for the VM with ID=1 via property name in the pipeline.

———————— EXAMPLE 8 ————————

PS C:>[PSCustomObject] @{ 'Name' = 'app1' } | Get-ArmorVM

Retrieves the details for the VM with Name='app1' via property name in the pipeline.

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

CHAPTER 13

Invoke Commands

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

13.1 Invoke-ArmorWebRequest

NAME Invoke-ArmorWebRequest

SYNOPSIS This cmdlet submits web requests to the Armor API.

SYNTAX Invoke-ArmorWebRequest [-Endpoint] <String> [[-Headers] <Hashtable>] [[-Method] <String>] [[-Body] <String>] [[-SuccessCode] <UInt16>] [[-Description] <String>] [<CommonParameters>]

DESCRIPTION This cmdlet sends custom HTTPS requests to the Armor API. It can be used for calling API endpoints that are not yet covered by the cmdlets in this module.

PARAMETERS

-Endpoint <String> Specifies the Armor API endpoint.

-Headers <Hashtable> Specifies the headers of the Armor API web request.

-Method <String> Specifies the method used for the Armor API web request. The permitted values are: - Delete - Get - Patch - Post - Put

-Body <String> Specifies the body of the Armor API web request. This parameter is ignored for Get requests.

-SuccessCode <UInt16> Specifies the value of the HTTP response code that indicates success for this Armor API web request.

-Description <String> If the PowerShell $ConfirmPreference value is elevated for this Armor API web request by setting the -Confirm parameter to $true, this specifies the text to display at the user prompt.

<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:>Invoke-ArmorWebRequest -Endpoint '/me'

Retrieves the current user's identity details.

———————— EXAMPLE 2 ————————

PS C:>Invoke-ArmorWebRequest -Endpoint '/vms' -Headers $Global:ArmorSession.Headers

Retrieves VM details using the session headers.

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

CHAPTER 14

Remove Commands

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

14.1 Remove-ArmorCompleteWorkload

NAME Remove-ArmorCompleteWorkload

SYNOPSIS This cmdlet deletes Armor Complete workloads.

SYNTAX Remove-ArmorCompleteWorkload [-ID] <UInt16> [[-ApiVersion] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION The specified workload in the Armor Complete account in context will be deleted if is empty.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete workload.

-ApiVersion <String> Specifies the API version for this request.

-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-ArmorCompleteWorkload -ID 1

If confirmed and empty of child objects, deletes workload with ID=1.

———————— EXAMPLE 2 ————————

PS C:>1 | Remove-ArmorCompleteWorkload

If confirmed and empty of child objects, deletes workload with ID=1 identified via pipeline value.

———————— EXAMPLE 3 ————————

PS C:>[PSCustomObject] @{ 'ID' = 1 | Remove-ArmorCompleteWorkload

If confirmed and empty of child objects, deletes workload with ID=1 identified via property name in the pipeline.

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

CHAPTER 15

Rename Commands

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

15.1 Rename-ArmorCompleteVM

NAME Rename-ArmorCompleteVM

SYNOPSIS Renames Armor Complete virtual machines.

SYNTAX Rename-ArmorCompleteVM [-ID] <UInt16> [-NewName] <String> [[-ApiVersion] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION The specified virtual machine in the Armor Complete account in context will be renamed.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete virtual machine that you want to rename.

-NewName <String> Specifies the new name for the Armor Complete virtual machine.

-ApiVersion <String> Specifies the API version for this request.

-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:>Rename-ArmorCompleteVM -ID 1 -NewName TEST-VM

Renames the VM with ID=1 to 'TEST-VM'.

———————— EXAMPLE 2 ————————

PS C:>[PSCustomObject] @{ 'ID' = 1; 'NewName' = 'TEST-VM' } | Rename-ArmorCompleteVM

Renames the VM with ID=1 to 'TEST-VM' via property names in the pipeline.

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

15.2 Rename-ArmorCompleteWorkload

NAME Rename-ArmorCompleteWorkload

SYNOPSIS Renames Armor Complete workloads.

SYNTAX Rename-ArmorCompleteWorkload [-ID] <UInt16> [-NewName] <String> [[-ApiVersion] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION The specified workload in the Armor Complete account in context will be renamed.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete workload that you want to rename.

-NewName <String> Specifies the new name of the Armor Complete workload.

-ApiVersion <String> Specifies the API version for this request.

-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:>Rename-ArmorCompleteWorkload -ID 1 -NewName TEST-WORKLOAD

Renames the workload with ID=1 to 'TEST-WORKLOAD'.

———————— EXAMPLE 2 ————————

PS C:>[PSCustomObject] @{ 'ID' = 1; 'NewName' = 'TEST-WORKLOAD' } | Rename-ArmorCompleteWorkload

Renames the workload with ID=1 to 'TEST-WORKLOAD' via property names in the pipeline.

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

CHAPTER 16

Reset Commands

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

16.1 Reset-ArmorCompleteVM

NAME Reset-ArmorCompleteVM

SYNOPSIS Resets Armor Complete virtual machines.

SYNTAX Reset-ArmorCompleteVM [-ID] <UInt16> [[-ApiVersion] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION The specified virtual machine in the Armor Complete account in context will be hard reset- effectively disconnecting the virtual power cord from the VM, plugging it back in, and then powering it back on. This reboot method has the potential to cause data corruption and should only be used when necessary.

See also: Restart-ArmorCompleteVM

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete virtual machine that you want to power off & on.

-ApiVersion <String> Specifies the API version for this request.

-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:>Reset-ArmorCompleteVM -ID 1

If confirmed, powers off & on the Armor Complete VM with ID=1.

———————— EXAMPLE 2 ————————

PS C:>1 | Reset-ArmorCompleteVM -Confirm:$false

Powers off & on the Armor Complete VM with ID=1 via pipeline value.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorVM -ID 1 | Reset-ArmorCompleteVM -Confirm:$false

Powers off & on the Armor Complete VM with ID=1 via property name in the pipeline without confirmation.

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

CHAPTER 17

Restart Commands

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

17.1 Restart-ArmorCompleteVM

NAME Restart-ArmorCompleteVM

SYNOPSIS Gracefully reboots virtual machines.

SYNTAX Restart-ArmorCompleteVM [-ID] <UInt16> [[-ApiVersion] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION The specified virtual machine will be gracefully rebooted in the Armor Complete account in context. VMware Tools or open-vm-tools must be installed and running for this request to succeed.

See also: Reset-ArmorCompleteVM

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete virtual machine that you want to gracefully reboot.

-ApiVersion <String> Specifies the API version for this request.

-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:>Retart-ArmorCompleteVM -ID 1

Gracefully reboot on the specified Armor Complete VM.

———————— EXAMPLE 2 ————————

PS C:>1 | Retart-ArmorCompleteVM

Reboot the Armor Complete VM with ID=1 specified via pipeline value.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorVM -ID 1 | Retart-ArmorCompleteVM

Reboot the Armor Complete VM with ID=1 via property name in the pipeline.

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

CHAPTER 18

Set Commands

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

18.1 Set-ArmorAccountContext

NAME Set-ArmorAccountContext

SYNOPSIS Sets the Armor Anywhere or Armor Complete account context.

SYNTAX Set-ArmorAccountContext [-ID] <UInt16> [<CommonParameters>]

DESCRIPTION If your user account has access to more than one Armor Anywhere and/or Armor Complete accounts, this cmdlet allows you to update the context, so that all future requests reference the specified account.

PARAMETERS

-ID <UInt16> Specifies which Armor account should be used for the context of all subsequent requests.

<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-ArmorAccountContext -ID 1

Set the account context to the specified account ID so that all subsequent commands reference that account.

———————— EXAMPLE 2 ————————

PS C:>2 | Set-ArmorAccountContext

Set the account context to 2 via the value in the pipeline.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorAccount -ID 3 | Set-ArmorAccountContext

Set the account context to 3 via the ID property name in the pipeline.

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

CHAPTER 19

Start Commands

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

19.1 Start-ArmorCompleteVM

NAME Start-ArmorCompleteVM

SYNOPSIS Starts Armor Complete virtual machines.

SYNTAX Start-ArmorCompleteVM [-ID] <UInt16> [[-ApiVersion] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION The specified virtual machine in the Armor Complete account in context will be powered on.

PARAMETERS

-ID <UInt16> Specifies the ID of the VM to power on in the Armor Complete account in context.

-ApiVersion <String> Specifies the API version for this request.

-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:>Start-ArmorCompleteVM -ID 1

Power on the Armor Complete VM with ID=1.

———————— EXAMPLE 2 ————————

PS C:>2 | Start-ArmorCompleteVM

Power on the Armor Complete VM with ID=2 via pipeline value.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorVM -ID 3 | Start-ArmorCompleteVM

Power on the Armor Complete VM with ID=3 via property name in the pipeline.

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

CHAPTER 20

Stop Commands

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

20.1 Stop-ArmorCompleteVM

NAME Stop-ArmorCompleteVM

SYNOPSIS Stops Armor Complete virtual machines.

SYNTAX Stop-ArmorCompleteVM [-ID] <UInt16> [[-Type] <String>] [[-ApiVersion] <String>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION The specified virtual machine in the Armor Complete account in context will be powered down.

PARAMETERS

-ID <UInt16> Specifies the ID of the Armor Complete virtual machine that you want to stop.

-Type <String> Specifies how you want to stop the Armor Complete virtual machine.

- Shutdown - Initiates a graceful shutdown of the operating system. - VMware Tools or open-vm-tools must be installed, running, and in a good

  state for this request to succeed.

  - This is the recommend way to stop your VMs.
- Poweroff - Initiates a hard shutdown of the VM- effectively disconnecting the virtual

  power cord from the VM.

  - This shutdown method has the potential to cause data corruption.
  - This should only be used when necessary.
- ForceOff - Breaks the state of the environment by marking the VM as powered off in

  the Armor Management Portal (AMP), but leaves the VM running in the Armor Complete cloud.

- This should not be used unless recommended by a Senior Armor Support team member.

-ApiVersion <String> Specifies the API version for this request.

-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:>Stop-ArmorCompleteVM -ID 1 -Type Shutdown

If confirmed, gracefully shutdown the specified Armor Complete VM.

———————— EXAMPLE 2 ————————

PS C:>2 | Stop-ArmorCompleteVM -Type Poweroff -Confirm:$false

Power off the Armor Complete VM with ID=2 via pipeline value without prompting for confirmation.

———————— EXAMPLE 3 ————————

PS C:>Get-ArmorVM -ID 3 | Stop-ArmorCompleteVM -Type ForceOff -Confirm:$false

Break the state of the Armor Complete VM with ID=3 via parameter name in the pipeline without prompting for confirmation, so that the VM appears to be powered off in the Armor Management Portal (AMP), but is still powered on in the Armor Complete cloud.

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