**DKC Cloud Code Base R1.0**

**Table of Contents**

Function AnsibleCreateLpgRouter 3

Function AnsibleCreateVcn 4

Function CheckSubnet 5

Function CheckVcn 6

Function Get-ChildCompartments 7

Function CopyBlockVolsInCompartment 8

Function CopyBootVolsInCompartment 9

Function GetActiveChildCompartment 10

Function GetActiveParentCompartment 12

Function GetBackupPolicies 13

Function GetBlockVolumes 14

Function GetBootVolumes 15

Function GetDbNodeName 16

Function GetDbSystems 17

Function GetLPGs 18

Function GetRouteTable 19

Function GetSubnet 20

Function GetTenantID 22

Function GetVcn 23

Function GetVM 25

Function GetVMs 27

Function GetVmNicAttachment 28

Function GetVmBootVolBackups 29

Function GetVmBlockVolBackups 31

Function GetBlockVolumes 33

Function ReadCsv 35

Function RestoreBootVol 36

Function ReturnValWithOptions 37

Function SelectBackupPolicy 38

Function SelectBlockVolume 40

Function SelectBootVolume 42

Function SelectLPG 43

Function SelectRouterTable 45

Function SelectSubnet 46

Function SelectVcn 48

JSON Input File tenant.json 50

Program BackupBlockVolumes.ps1 52

Program BackupBootVolumes.ps1 53

Program GetActiveChildCompartment.ps1 54

Program GetActiveParentCompartment.ps1 55

Program GetBackupPolicy.ps1 56

Program GetDbNode.ps1 57

Program GetLpg.ps1 58

Program GetRouterTable.ps1 59

Program GetSubnet.ps1 60

Program GetVcn.ps1 61

Program GetVm.ps1 62

Program GetVmBlockVol.ps1 63

Program GetVmBlockVolBkup.ps1 64

Program GetVmBootVol.ps1 65

Program GetVmBootVolBkup.ps1 66

Program StartVMs.ps1 67

Program StopVMs.ps1 68

Program TestRestoreVmFromBkup.ps1 69

# Function AnsibleCreateLpgRouter

**Synopsis:**

This function creates an a shell script that creates OCI routing tables for local peering gateways. The shell script calls the Ansible script 014\_CreateLpgRouter.Yaml. 014\_CreateLpgRouter.Yaml calls the OCI APIs to create the resource.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myLpgRouter | string | Must be the name of the LPG router that will be created |
| myCompartment | array | Dictionary object created by function Get-ChildCompartments |
| myVcn | array | Dictionary object created by function GetVcn |
| myLpg | array | Dictionary object created by function GetLPG |

**Examples:**

AnsibleCreateLpgRouter myRouterTableName $Compartment $Vcn $Lpg

**Return Values: None**

# Function AnsibleCreateVcn

**Synopsis:**

This function creates an a shell script that creates OCI virtual cloud networks (VCNs). The shell script calls the Ansible script 010\_PrepareToCreateVCN.yaml. 010\_PrepareToCreateVCN.yaml calls the OCI APIs to create the resource. The name of the VCN is derived from the compartment name. For example, of the compartment name is automation then the name of the VCN shall be automation\_vcn

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | string | The name of the compartment where the VCN is to be created |
| myCompaertmentID | string | The OCID of the compartment |
| myVcnCidr | string | The CIDR of the VCN |

**Examples:**

AnsibleCreateVcn automation <compartment OCID> 10.0.0.0/24

**Return Values: None**

# Function CheckSubnet

**Synopsis:**

This function checks for a virtual cloud network subnet.If the subnet exists, it returns a null value. If the subnet does not exist, it creates the subnet for the VCN. This function is called by ManageOciNetworkInfrastructure.ps1

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompaertmentID | string | The OCID of the compartment |
| myCidr | string | The CIDR of the subnet |

**Examples:**

CheckSubnet $myCompartment $myCidr

**Return Values: None**

# Function CheckVcn

**Synopsis:**

This function checks for a virtual cloud network (VCN) . If the VCN exists, it returns a null value. If the VCN does not exist, it creates it. This function is called by ManageOciNetworkInfrastructure.ps1

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompaertmentID | string | The OCID of the compartment |
| myCidr | string | The CIDR of the VCN |

**Examples:**

CheckVcn $myCompartment $myCidr

**Return Values: None**

# Function Get-ChildCompartments

**Synopsis:**

This function checks for and returns all of the properties of all child compartments of the specified parent compartment OCID. Returns a null value on failure

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myParentCompartmentId | string | The OCID of the parent compartment |

**Examples:**

**Return Values:**

Get-ChildCompartments <OCID of parent compartment>

| Key | Returned | Description |
| --- | --- | --- |
| data | int | The numeric value of the compartment object returned by the OCI API |
| data.freeform-tags | array | Array of strings for any freeform tags created with the compartment |
| [data.is](http://data.is)-accessible | null |  |
| [data.id](http://data.id) | string | The OCID of the child compartment |
| data.time-created | date | The date stamp when the compartment was created |
| data.description | string | The descriptive name given to the compartment when it was created |
| data.defined-tags | array | Array of strings of defined tags that are associated with the compartment |
| data.name | string | The name of the compartment |
| data.lifecycle-state | string | The state of the compartment, typically is ACTIVE, but may have other states. See <https://docs.cloud.oracle.com/en-us/iaas/api/#/en/identity/20160918/Compartment/> |
| data.inactive-status | null |  |
| data.compartment-id | string | The OCID of the parent compartment |

# Function CopyBlockVolsInCompartment

**Synopsis:**

This complex function copies all block volume backup copies from the specified compartment to the specified data center region. The tenant must be subscribed to the OCI region to which you wish to copy backups. See managing regions at <https://docs.cloud.oracle.com/en-us/iaas/Content/Identity/Tasks/managingregions.htm> and ensure your tenant is properly configured prior to calling this function

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | array | The dictionary object of the compartment as returned by the function GetActiveChildCompartment |
| myTargetRegion | string | The name of the OCI data center region where you want to copy the block volume backups |

**Examples:**

CopyBlockVolsInCompartment $myCompartment “us-phoenix-1”

**Return Values:**

The function returns standard output returned by the OCI API function CopyVolumeBackup. See <https://docs.cloud.oracle.com/en-us/iaas/api/#/en/iaas/20160918/VolumeBackup/CopyVolumeBackup> and <https://docs.cloud.oracle.com/en-us/iaas/tools/oci-cli/2.8.2/oci_cli_docs/cmdref/bv/backup/copy.html>

# Function CopyBootVolsInCompartment

**Synopsis:**

This complex function copies all boot volume backup copies from the specified compartment to the specified data center region. The tenant must be subscribed to the OCI region to which you wish to copy backups. See managing regions at <https://docs.cloud.oracle.com/en-us/iaas/Content/Identity/Tasks/managingregions.htm> and ensure your tenant is properly configured prior to calling this function

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | array | The dictionary object of the compartment as returned by the function GetActiveChildCompartment |
| myTargetRegion | string | The name of the OCI data center region where you want to copy the block volume backups |

**Examples:**

CopyBootVolsInCompartment $myCompartment “us-phoenix-1”

**Return Values:**

The function returns standard output returned by the OCI API function CopyVolumeBackup. See https://docs.cloud.oracle.com/en-us/iaas/api/#/en/iaas/20160918/BootVolumeBackup/CopyBootVolumeBackup and <https://docs.cloud.oracle.com/en-us/iaas/tools/oci-cli/2.8.2/oci_cli_docs/cmdref/bv/boot-volume-backup/copy.html>

# Function GetActiveChildCompartment

**Synopsis:**

This function checks for and returns properties of the active child compartment if found. Otherwise, the return value is null.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartmentID | array | The dictionary object of all child compartments as created by Get-ChildCompartments |
| myCompartment | string | The name of the active child compartment to search for |

**Examples:**

GetActiveChildCompartment $myCompartmentID $myCompartment

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| freeform-tags | array | Array of strings for any freeform tags created with the compartment |
| Is-accessible | null |  |
| id | string | The OCID of the child compartment |
| time-created | date | The date stamp when the compartment was created |
| description | string | The descriptive name given to the compartment when it was created |
| defined-tags | array | Array of strings of defined tags that are associated with the compartment |
| name | string | The name of the compartment |
| lifecycle-state | string | The state of the compartment, typically is ACTIVE, but may have other states. See <https://docs.cloud.oracle.com/en-us/iaas/api/#/en/identity/20160918/Compartment/> |
| inactive-status | null |  |
| compartment-id | string | The OCID of the parent compartment |

# Function GetActiveParentCompartment

**Synopsis:**

This function checks for and returns properties of the active parent compartment if found. Otherwise, the return value is null.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartmentID | array | The dictionary object the parent compartments |
| myCompartment | string | The name of the active parent compartment to search for |

**Examples:**

GetActiveChildCompartment $myCompartmentID $myCompartment

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| freeform-tags | array | Array of strings for any freeform tags created with the compartment |
| Is-accessible | null |  |
| id | string | The OCID of the child compartment |
| time-created | date | The date stamp when the compartment was created |
| description | string | The descriptive name given to the compartment when it was created |
| defined-tags | array | Array of strings of defined tags that are associated with the compartment |
| name | string | The name of the compartment |
| lifecycle-state | string | The state of the compartment, typically is ACTIVE, but may have other states. See <https://docs.cloud.oracle.com/en-us/iaas/api/#/en/identity/20160918/Compartment/> |
| inactive-status | null |  |
| compartment-id | string | The OCID of the parent compartment |

# Function GetBackupPolicies

**Synopsis:**

This function checks for and returns properties of all policies in the specified compartment

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | array | The dictionary object of all child compartments as created by Get-ChildCompartments |

**Examples:**

GetBackupPolicies $myCompartment

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of backup policy objects return from the API |
| data.compartment-id | string | The OCID of the compartment that contains the backup policies |
| data.display-name | string | The name of the policy |
| [data.id](http://data.id) | string | The OCID of the policy |
| data.schedules | array | An array of objects that hold the schedules of the specified policy. See https://docs.cloud.oracle.com/en-us/iaas/api/#/en/iaas/20160918/VolumeBackupPolicy/ |

# Function GetBlockVolumes

**Synopsis:**

This function checks for and returns properties of all block volumes in the availability domain that the VM resides in

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVM | array | The dictionary object of the VM as returned by GetVM |

**Examples:**

GetBlockVolumes $myVM

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of block volumes objects return from the API |
| data.availability-domain | string | The name of the region availability domain where the block volumes reside |
| data.compartment-id | string | The OCID of the compartment that contains the block volumes |
| data.display-name | string | The name of the block volume |
| data.instance-id | string | The OCID of the instance to which the block volume is attached |
| data.lifecycle-state | string | The state of the device. The most common type is “ATTACHED”. See <https://docs.cloud.oracle.com/en-us/iaas/api/#/en/iaas/20160918/Volume/> |
| data.time-created | date | The date the block volume was created |

# Function GetBootVolumes

**Synopsis:**

This function checks for and returns properties of all boot volumes in the availability domain that the VM resides in

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVM | array | The dictionary object of the VM as returned by GetVM |

**Examples:**

GetBootVolumes $myVM

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of boot volumes objects return from the API |
| data.availability-domain | string | The name of the region availability domain where the boot volumes reside |
| data.compartment-id | string | The OCID of the compartment that contains the boot volumes |
| data.display-name | string | The name of the boot volume |
| data.instance-id | string | The OCID of the instance to which the boot volume is attached |
| data.lifecycle-state | string | The state of the device. The most common type is “ATTACHED”. See https://docs.cloud.oracle.com/en-us/iaas/api/#/en/iaas/20160918/BootVolume/ |
| data.time-created | date | The date the boot volume was created |

# Function GetDbNodeName

**Synopsis:**

This function checks for and returns properties of a virtual machine database service node

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myDbSystems | array | The dictionary object of the database systems as returned by the function GetDbSystems |
| myDbNodeName | string | The name of the service node to search for |

**Examples:**

GetDbNodeName $myDbSystems $myDbNodeName

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| fault-domain | string | The fault domain where the service node resides |
| software-storage-size-in-gb | int | The size in GB of the service node’s boot volume |
| hostname | string | The name of the service node |
| time-created | date | The date the service node was created |
| lifecycle-state | string | The state of the service node. States may be AVAILABLE, STOPPED, STARTING, STOPPING, TERMINATING, TERMINATED |
| vnic-id | string | The OCID of the VNIC assigned to the service node |
| id | string | The OCID of the service node |
| db-system-id | string | The OCID of the DBaaS system to which this service node is assigned to |
| backup-vnic-id | string | If applicable, the OCID of the backup VNIC for this service node |

# Function GetDbSystems

**Synopsis:**

This function checks for and returns properties of all virtual machine database system within a compartment

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | array | The dictionary object of the compartment as returned by the function GetActiveChildCompartment |

**Examples:**

GetDbSystems $myCompartment

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Array of objects that describe all DBaaS VMs found within the compartment |
| data.availability-domain | string | The availability domain of the DBaaS system |
| boot-volume-id | string | The OCID of the Oracle database compartment that holds the database |
| data.compartment-id | string | The OCID of the compartment that holds the DBaaS |
| data.display-name | string | The name of the DBaaS resource |
| [data.id](http://data.id) | string | The OCID of the DBaaS resource |
| data.instance-id | string | Same as [data.id](http://data.id) |
| data.lifecycle-state | string | The state of the compartment. Can be “ATTACHED, DETACHED, TERMINATED” |
| data.time-created | date | The date that the resource was created |

# Function GetLPGs

**Synopsis:**

This function checks for and returns properties of all local peering gateways that are members of a VCN. It returns a dictionary object containing the properties of the LPGs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVCN | array | The dictionary object of the VCN that holds the LPGs |

**Examples:**

GetLPGs $myVCN

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Array of objects that describe all LPGs associated with the VCN |
| route-table-id | string | The OCID of the route table the LPG is assigned to |
| vcn-id | string | The OCID that the LPG is associated with |
| is-cross-tenancy-peering | string | The status of cross tenancy peering. Values are True or False |
| data.display-name | string | The name of the DBaaS resource |
| lifecycle-state | string | The life cycle status of the LPG. |
| compartment-id | string | The OCID of the compartment to which the LPG is a member of |
| time-created | date | The date that the resource was created |
| display-name | string | The display name of the LPG |
| peer-advertised-cidr-details | array | The CIDRs of the peered VCNs by this LPG. Values within the array elements are strings |
| peering-status | string | The status of the LPG’s peering |
| peer-advertised-cidr | string | The route that the LPG advertises to the VCN |

# Function GetRouteTable

**Synopsis:**

This function checks for and returns properties of all route tables that are members of a VCN. It returns a dictionary object containing the properties of the route tables.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVCN | array | The dictionary object of the VCN that holds the LPGs |

**Examples:**

GetRouteTable $myVCN

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Array of objects that describe all route tables associated with the VCN |
| vcn-id | string | The OCID of the VCN that the route table is a member of |
| lifecycle-state | string | The life cycle status of the route table |
| display-name | string | The display name of the route table |
| time-created | date | The date stamp when the resource was completed |
| route-rules | array | An array of route rules built for this route table |
| route-rules.network-entity-id | string | The OCID of the LPG to send traffic to |
| route-rulesdestination-type | string | The destination type for the route, usually CIDR\_BLOCK |
| route-rules.destination | string | The destination CIDR |
| compartment-id | string | The compartment OCID where the route table resides |
| id | string | The OCID of the route table |

# Function GetSubnet

**Synopsis:**

This function checks for and returns properties of all subnets that are members of a VCN. It returns a dictionary object containing the properties of the subnets within the VCN.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVCN | array | The dictionary object of the VCN that holds the LPGs |

**Examples:**

GetSubnet $myVCN

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Array of objects that describe all subnets associated with the VCN |
| ipv6-cidr-block | string | The IP V6 CIDR assigned to the subnet |
| virtual-router-mac | string | The MAC address assigned to the VCN |
| compartment-id | string | The compartment OCID where the subnet resides |
| security-list-ids | array | The security list(s) applied to this subnet |
| dns-label | string | The subnet label assigned to the subnet |
| subnet-domain-name | string | The FQDN of the subnet |
| dhcp-options-id | string | The OCID of the DHCP object that defines DHCP options to the subnet |
| vcn-id | string | The OCID of the VCN to which the subnet is a member of |
| virtual-router-ip | string | The IP address assigned to the router for this subnet |
| display-name | string | The display name of the subnet |
| cidr-block | string | The CIDR block assigned to the subnet |
| lifecycle-state | string | The life cycle status of the subnet |
| availability-domain | string | The availability domain to which the subnet is applied. This is not used in DKC cloud deployments since all network resources in our builds are built across availability domains |
| ipv6-virtual-router-ip | string | The IPv6 address assigned to the subnet if IPv6 addresses are used |
| route-table-id | string | The OCID of the route table assigned to the subnet |
| time-created | date | The date the resource was created |
| prohibit-public-ip-on-vnic | boolean | Determines if public IPs can be assigned to resources within the subnet. Valid values are True or False |
| ipv6-public-cidr-block | string | The IPv6 CIDR assigned to the subnet if IPv6 addressing is used |
| id | string | The OCID of the subnet |

# Function GetTenantID

**Synopsis:**

This function checks for and returns properties regarding the OCI subscription tenancy

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myTenantId | string | The OCID of the tenancy |

**Examples:**

GetTenantID $myTenantId

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of tenancy objects return from the API |
| compartment-id | string | null |
| description | string | The description of the tenancy |
| id | string | The OCID of the tenancy |
| inactive-status | string | Should always be null |
| is-accessible | boolean | Values are True or False |
| lifecycle-state | string | should alwys be AVAILABLE |
| name | string | The display name of the tenancy |
| time-created | date | The date the tenancy was created |

# Function GetVcn

**Synopsis:**

This function checks for and returns properties regarding the VCNs within the specified compartment

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | array | A dictionary object that describes the compartment where the VCNs reside |

**Examples:**

GetVCN $myCompartment

**Return Values: (in JSON)**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of VCN objects return from the API |
| cidr-block | string | The CIDR of the VCN |
| compartment-id | string | The compartment OCID where the VCN resides |
| default-dhcp-options-id | string | The OCID of the DHCP object that defines DHCP options to the subnet |
| default-route-table-id | string | The default OCID of the router table applied to any subnet members of the VCN that do not have a defined route table |
| default-security-list-id | string | The default OCID of the security list applied to any subnet members of the VCN that do not have a defined security list |
| display-name | string | The display name of the VCN |
| dns-label | string | The DNS label for the VCN |
| id | string | The OCID of the VCN |
| ipv6-cidr-block | string | The IPv6 CIDR block if IPv6 addresses are used |
| ipv6-public-cidr-block | string | The IPv6 CIDR block if public IPv6 addresses are used |
| lifecycle-state | string | The life cycle state of the VCN |
| time-created | date | The date stamp when the VCN object was created |
| vcn-domain-name | string | The FQDN name for the VCN object |

# Function GetVM

**Synopsis:**

This function checks for and returns properties regarding the OCI VM specified in the argument vector. Function requires the dictionary object created by GetVMs be passed to it in addition to the string value of the VM name

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVMs | array | Dictionary object of VMs within a compartment |
| myVmName | string | The display name of the VM to seaerch for |

**Examples:**

GetVM $myVms $myVmName

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| launch-options | array | Options set for the launch of this VM |
| [launch-options.is](http://launch-options.is)-pv-encryption-in-transit-enabled | boolean | Should always be false for DKC deployments |
| launch-options.network-type | string | Should always be PARAVIRTUALIZED for DKC deployments |
| launch-options.boot-volume-type | string | Should always be PARAVIRTUALIZED for DKC deployments |
| [launch-options.is](http://launch-options.is)-consistent-volume-naming-eabled | boolean |  |
| launch-options.firmware | string | The cloud physical machine firmware applied to the paravirtualized VM |
| launch-options.remote-data-volume-type | string | Should always be PARAVIRTUALIZED for DKC deployments |
| launch-mode | string | Should always be PARAVIRTUALIZED for DKC deployments |
| metadata | array | An array of strings. The array is formatted using the python function open(with ssh\_keys, …..) |
| time-created | date | The date the VM object was created |
| source-details | array | An array of strings that keeps a history of the original image source |
| source-details.source-type | string | Is usually “image” |
| source-details.kms-key-id | string | Usually null |
| source-details.boot-volume-size-in-gbs | string | Usually null |
| source-details.image-id | string | The OCID of the original source image that was used to create the boot volume. This OCID should not be relied upon for cloud automation |
|  |  |  |

# Function GetVMs

**Synopsis:**

This function checks for and returns properties regarding the VMs within a compartment

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | Array | Dictionary object created by GetActiveChildCompartment containing properties describing the compartment VMs are a member of |

**Examples:**

GetVMs $myCompartment

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of VM objects return from the API |
| All other objects |  | Array elements for each VM as described above in GetVM |

# Function GetVmNicAttachment

**Synopsis:**

This function checks for and returns properties regarding the VNIC object

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVM | Array | Dictionary object created by GetVM containing properties describing a VM |

**Examples:**

GetVmNicAttachment $myVM

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| lifecycle-state | string | State of the NIC, should always be ATTACHED |
| time-created | date | Date the object was created |
| display-name | string | optional, usually null |
| compartment-id | string | The compartment OCID where the NIC is located at |
| nic-index | int | Index number for the VNIC, usually 0 |
| vlan-tag | int | The availability domain VLAN number assigned to the VNIC |
| instance-id | string | The VM instance the VNIC is attached to |
| id | string | The OCID of the VNIC |
| subnet-id | string | The OCID of the subnet the VNIC is bound to |
| availability-domain | string | The name of the availability domain the VNIC is a member of |

# Function GetVmBootVolBackups

**Synopsis:**

This function checks for and returns properties regarding all VM Boot Volume backup objects within the specified compartment. This includes all backup objects in all availability domains and regions that the compartment spans.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | Array | Dictionary object created by GetActiveChildCompartment containing properties describing the compartment |

**Examples:**

GetVmBootVolBackups $myCompartment

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of VM boot volume backup objects return from the API |
| source-type | string | Describes how the object was created. Usually SCHEDULED |
| time-created | date | Data stamp of when the object was created |
| unique-size-in-gbs | int | Actual storage of the backup |
| size-in-gb | int | The size of the volume if restored from this and other volume backups |
| type | string | Is either INCREMENTAL or FULL |
| time-request-received | date | Date stamp when the backup was requested |
| display-name | string | Name of the backup object, either as defined by a backup policy schedule or as defined by the user if a manual backup |
| id | string | The OCID of the backup object |
| boot-volume-id | string | The OCID of the original volume from which the backup was created |
| compartment-id | string | The compartment the objects are a member of |
| expiration-time | date | The date and time that the backup object will be terminated |
| lifecycle-state | string | The state of the object, usually AVAILABLE or TERMINATED |
| image-id | string | The original image OCID from which the boot volume had been created from |

# Function GetVmBlockVolBackups

**Synopsis:**

This function checks for and returns properties regarding all VM Block Volume backup objects within the specified compartment. This includes all backup objects in all availability domains and regions that the compartment spans.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myCompartment | Array | Dictionary object created by GetActiveChildCompartment containing properties describing the compartment |

**Examples:**

GetVmBlockVolBackups $myCompartment

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of VM boot volume backup objects return from the API |
| source-type | string | Describes how the object was created. Usually SCHEDULED |
| time-created | date | Data stamp of when the object was created |
| unique-size-in-gbs | int | Actual storage of the backup |
| size-in-gb | int | The size of the volume if restored from this and other volume backups |
| type | string | Is either INCREMENTAL or FULL |
| time-request-received | date | Date stamp when the backup was requested |
| display-name | string | Name of the backup object, either as defined by a backup policy schedule or as defined by the user if a manual backup |
| id | string | The OCID of the backup object |
| boot-volume-id | string | The OCID of the original volume from which the backup was created |
| compartment-id | string | The compartment the objects are a member of |
| expiration-time | date | The date and time that the backup object will be terminated |
| lifecycle-state | string | The state of the object, usually AVAILABLE or TERMINATED |
| image-id | string | The original image OCID from which the boot volume had been created from |

# Function GetBlockVolumes

**Synopsis:**

This function checks for and returns properties regarding block volume objects that are attached to a VM

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| myVM | Array | Dictionary object created by GetVM containing properties describing the VM |

**Examples:**

GetBlockVolumes $myVM

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| data | array | Integer values that represent the number of block volumes objects return from the API |
| is-read-only | boolean | Always False with DKC deployments |
| availability-domain | string | The name of the availability domain where the block volume is located |
| time-created | date | The date stamp the object was created |
| attachment-type | string | Usually paravirtualized but sometimes iscsi |
| id | string | The OCID of the block volume attachment |
| lifecycle-state | string | Always ATTACHED if the block volume is attached to a VM, but may have other states per <https://docs.cloud.oracle.com/en-us/iaas/api/#/en/iaas/20160918/Volume/ListVolumes> |
| volume-id | string | The OCID of the block volume |
| is-shareable | boolean | Ether True or False |
| instance-id | string | The OCID to which the block volume is attached |
| is-pv-encryption-in-transit-enabled | boolean | Usually False |
| device | string | The device identity on the VM |
| compartment-id | string | The OCID of the compartment the object is a member of |
| display-name | string | The name of the object |

# Function ReadCsv

**Synopsis:**

This function reads a CSV file and returns a dictionary object containing data read from the file. It requires that the CSV file have a semicolon delimiter. We use this function in the codebase when tenants are initially built out

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myFile | string | The fully qualified path where the CSV file is located |

**Examples:**

ReadCsv $myFile

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| Various | array | Array containing objects of strings or integers up to 16 columns in length |

# Function RestoreBootVol

**Synopsis:**

This function restores a boot volume from the specified backup object $myVolToRestore to the specified volume name $myNewVmName using the properties of the source VM $myVmName. The function makes no assumption that the specified source VM and source boot volume backup exist and will through an exception if said objects are not found. The return value is either an error code or a JSON object representation of the restored volume. The most recent backup is restored. See <https://docs.cloud.oracle.com/en-us/iaas/api/#/en/iaas/20160918/BootVolume/CreateBootVolume> for more information regarding the REST API called by this function.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myVolToRestore | array | A dictionary object containing properties of the selected boot volume to restore |
| $myVmName | array | A dictionary object containing properties of the source VM. This is used to determine the properties of the source VM |
| $myNewVmName | array | A dictionary object that describes the properties of the target object to restore |

**Examples:**

RestoreBootVol $myVolToRestore $myVmName $myNewVmName

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| Various | JSON | See referenced API link above for details |

# Function ReturnValWithOptions

**Synopsis:**

This function returns the requested type of data from the dictionary object passed to it

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myProgramName | string | The name of the program calling this function |
| $myReturnValue | array | The dictionary object the function parses |
| $myOption | string | The dictionary key-pair to return to the calling program. Values may be ALL, BLOCKVOLID, BOOTVOLID, COMPARTMENT, DISPLAYNAME, OCID |

**Examples:**

ReturnDataValWithOptions $myProgramName $myReturnValue $myOptions

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| Various | Dictionary object or string | Return value depends on how the function is called |

# Function SelectBackupPolicy

**Synopsis:**

This function selects and returns backup policy $myBackupPolicy from the dictionary object $myBackupPolicies.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myBackupPolicies | array | Dictionary object created from GetBackupPolicies |
| $myBackupPolicy | string | String value containing name of backup policy to return |

**Examples:**

SelectBackupPolicy $myBackupPolicies $myBackupPolicy

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| time-created | date | Date the object was created |
| display-name | string | The name of the backup policy |
| id | string | The OCID of the backup policy |
| compartment-id | string | The OCID the object is a member of |
| schedules | array | The schedules associated with the backup policy |
| schedules.month | string | Usually JANUARY, unless schedule is defined to run 1 time per month |
| schedules.day-of-month | int | usually 1, unless schedule is defined to run on a day of the month |
| schedules.backup-type | string | INCREMENTAL or FULL |
| offset-type | string | Always STRUCTURED |
| day-of-week | string | Varies |
| period | string | Always ONE\_DAY |
| retention-seconds | int | Number of seconds to retain the backup |
| time-zone | string | Should always be set to “REGIONAL\_DATA\_CENTER\_TIME” |
| hour-of-day | int | The hour to start the backup |

# Function SelectBlockVolume

**Synopsis:**

This function returns all block volumes from $myBlockVolumes that match $myVM

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myBlockVolumes | array | Dictionary object of block volumes created by GetBlockVolumes |
| $myVM | array | Dictionary object describing a VM as created by GetVM |

**Examples:**

SelectBlockVolume $myBlockVolumes $myVM

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| id | string | OCID of volume ID attachment |
| display-name | string | Name of the block volume |
| time-created | date | Date the object was created |
| volume-id | string | OCID of the block volume |
| availability-domain | string | Name of the availability domain where the block volume was created |
| is-pv-encryption-in-transit-enabled | boolean | Always False |
| device | string | Name of device as presented to VM |
| is-read-only | boolean | Always False |
| compartment-id | string | OCID of the compartment the object is a member of |
| lifecycle-state | string | Should always be ATTACHED when volume is attached to a a VM |
| is-shareable | boolean | True or False |
| attachment-type | string | Usually “paravirtualized”, but may be “iscsi” |
| instance-id | string | OCID of the VM the volume is attached to |

# Function SelectBootVolume

**Synopsis:**

This function returns all boot volumes from $myBlockVolumes that match $myVM

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myBootVolumes | array | Dictionary object of boot volumes created by GetBlockVolumes |
| $myVM | array | Dictionary object describing a VM as created by GetVM |

**Examples:**

SelectBootVolume $myVM $myBlockVolumes

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| id | string | OCID of the instance the boot volume is attached to |
| display-name | string | Usually “Remote boot attachment for instance” |
| lifecycle-state | string | Always ATTACHED |
| time-created | date | Date the object was created |
| is-pv-encryption-in-transit-enabled | boolean | Always False |
| boot-volume-id | string | OCID of the boot volume object |
| compartment-id | string | OCID of the compartment the object is a member of |
| instance-id | string | OCID of the VM the boot volume is attached to |
| availability-domain | string | Availability domain where the object exists |

# Function SelectLPG

**Synopsis:**

This function returns the dictionary object from $myLPGs that matches the string value of $$myLpgName

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myLpgName | string | Name of the LPG to search for |
| $myLPGs | array | Dictionary object describing all LPGs within a compartment |

**Examples:**

SelectLPG $myLpgName $myLPGs

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| time-created | date | Date the object was created |
| display-name | string | Name of the LPG |
| peering-status | string | Should be “PEERED” when LPG has been peered with another LPG. WARNING! LPGs that were formerly peered cannot be peered again. The formerly peered LPG must be terminated and then recreated to be re-peered |
| is-cross-tenancy-peering | boolean | Always False in DKC deployments |
| vcn-id | string | OCID of the VCN the LPG is a member of |
| id | string | OCID of the LPG object |
| peer-advertised-cidr-details | array | CIDR routes the LPG will advertise |
| peering-status-details | string | Always “Connected to a peer.” |
| compartment-id | string | OCID of the compartment the object is a member of |
| lifecycle-state | string | Should always be AVAILABLE |
| peer-advertised-cidr | string | The route advertised by the remote LPG this LPG is peered with |
| route-table-id | string | The OCID of the route table the LPG is associated with, null if not associated with a router table |

# Function SelectRouterTable

**Synopsis:**

This function returns the route table dictionary object that matches $myRouterTableName from $myRouterTables

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myRouterTableName | string | Name of the router table to search for |
| $myRouteTables | array | Dictionary object describing all router tables that are associated with a VCN |

**Examples:**

SelectRouterTable $myRouterTableName $myRouteTables

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| display-name | string | Name of the route table object |
| time-created | date | Date the object was created |
| route-table-rules | array | Array contains route table rules built for the route table object |
| compartment-id | string | OCID of the compartment the object is a member of |
| id | string | OCID of the route table object |
| lifecycle-state | string | Should always be AVAILABLE |
| vcn-id | string | OCID of the VCN the route table is a member of |

# Function SelectSubnet

**Synopsis:**

This function returns the subnet dictionary object that matches $mySubnetName from $mySubnets

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $mySubnetName | string | Name of the subnet to search for |
| $mySubnets | array | Dictionary object describing all subnets that are associated with a VCN |

**Examples:**

SelectSubnet $mySubnets $mySubnetName

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| ipv6-cidr-block | string | The IP V6 CIDR assigned to the subnet |
| virtual-router-mac | string | The MAC address assigned to the VCN |
| compartment-id | string | The compartment OCID where the subnet resides |
| security-list-ids | array | The security list(s) applied to this subnet |
| dns-label | string | The subnet label assigned to the subnet |
| subnet-domain-name | string | The FQDN of the subnet |
| dhcp-options-id | string | The OCID of the DHCP object that defines DHCP options to the subnet |
| vcn-id | string | The OCID of the VCN to which the subnet is a member of |
| virtual-router-ip | string | The IP address assigned to the router for this subnet |
| display-name | string | The display name of the subnet |
| cidr-block | string | The CIDR block assigned to the subnet |
| lifecycle-state | string | The life cycle status of the subnet |
| availability-domain | string | The availability domain to which the subnet is applied. This is not used in DKC cloud deployments since all network resources in our builds are built across availability domains |
| ipv6-virtual-router-ip | string | The IPv6 address assigned to the subnet if IPv6 addresses are used |
| route-table-id | string | The OCID of the route table assigned to the subnet |
| time-created | date | The date the resource was created |
| prohibit-public-ip-on-vnic | boolean | Determines if public IPs can be assigned to resources within the subnet. Valid values are True or False |
| ipv6-public-cidr-block | string | The IPv6 CIDR assigned to the subnet if IPv6 addressing is used |
| id | string | The OCID of the subnet |

# Function SelectVcn

**Synopsis:**

This function returns the subnet dictionary object that matches $myVcnName from $myVCNs

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $myVCNName | string | Name of the VCN to search for |
| $myVCNs | array | Dictionary object describing all VCNs in a compartment |

**Examples:**

SelectVcn $myVCNs $myVcnName

**Return Values:**

| Key | Returned | Description |
| --- | --- | --- |
| cidr-block | string | The CIDR of the VCN |
| compartment-id | string | The compartment OCID where the VCN resides |
| default-dhcp-options-id | string | The OCID of the DHCP object that defines DHCP options to the subnet |
| default-route-table-id | string | The default OCID of the router table applied to any subnet members of the VCN that do not have a defined route table |
| default-security-list-id | string | The default OCID of the security list applied to any subnet members of the VCN that do not have a defined security list |
| display-name | string | The display name of the VCN |
| dns-label | string | The DNS label for the VCN |
| id | string | The OCID of the VCN |
| ipv6-cidr-block | string | The IPv6 CIDR block if IPv6 addresses are used |
| ipv6-public-cidr-block | string | The IPv6 CIDR block if public IPv6 addresses are used |
| lifecycle-state | string | The life cycle state of the VCN |
| time-created | date | The date stamp when the VCN object was created |
| vcn-domain-name | string | The FQDN name for the VCN object |

# JSON Input File tenant.json

**Synopsis:**

This JSON text readable file contains all required input data for the DKC codebase. The file is read by all Powershell programs and is converted to a dictionary object in hash table format. The file contains sensitive information and must be kept highly secure.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| ClientName | string | Name of the institution |
| ClientShortName | string | Short name used in the OCI tenant build |
| TenantId | string | The OCID, or root compartment ID, of the tenant |
| ParentCompartmentName | string | The name, in CAPS, of the parent compartment. This parent compartment is created as a child of the root compartment. The entire codebase is dependent on the parent/child compartment hierarchy. |
| BackupCompartmentName | string | This is the name of the compartment that will hold all backup policies for the tenant. Only backup objects are to be stored within this compartment. No other objects of any kind are to be stored in this compartment. |
| BastionCompartmentName | string | This is the name of the compartment that will hold objects that provide “jump host” or “bastion” services in accordance with best practices for modern cloud framework architectures. Examples of bastion objects include a VCN, a subnet, route tables, an internet gateway router, peering routers, and VMs that provide either remote desktop services or SSH services into the tenant. Access is permitted only in the event that the institution declares a disaster or if the IPSEC tunnel that provides service to the tenant is inoperative. |
| BastionVcnCidr | string | The CIDR of the VCN and subnet that is to be created within the compartment. The CIDR of the subnet will match the CIDR of the VCN. |
| BastionRouteTableCsvFle | string | Not used in this release of the codebase. |
| DatabaseCompartmentName | string | The name of the compartment that will hold all objects that are databases, or objects that support database operations, such as RMAN, disk SNAP backups, and NFS shares. No mid or web tier or any other non database objects are to be created within this compartment. Access to this compartment’s resources must be highly restricted. The database compartment hosts the institution’s most sensitive data. |
| DatabaseVcnCidr | string | The CIDR of the VCN and subnet that is to be created within the compartment. The CIDR of the subnet will match the CIDR of the VCN. |
| DmzCompartmentName | string | Not used in this release of the codebase. |
| IntranetCompartmentName | string | Not used in this release of the codebase. |
| IntranetVcnCidr | string | Not used in this release of the codebase. |
| TestCompartmentName | string | The name of the compartment that will host all test mid and web tier hosts. No production objects are to be hosted in this compartment. |
| TestVcnCidr | string | The CIDR of the VCN and subnet that is to be created within the compartment. The CIDR of the subnet will match the CIDR of the VCN. |
| VpnCompartmentName | string | The name of the compartment that hosts site-to-site VPN cloud objects that support the tenant’s primary region. Only cloud VPN objects are to be stored within this compartment. All VCN and VPN objects within the compartment are manually created, as is all routing. These connections are to remain nailed up. |
| WebCompartmentName | string | The name of the compartment that will host all production mid and web tier applications as well as NLB and WAF appliances. Only production cloud objects are to be hosted in this compartment. |
| WebVcnCidr | string | The CIDR of the web compartment. This must be a /23 CIDR that will be assigned to the VCN. The CIDR will be subnetted into a pair of /25 subnets in the first /24 of the address space and will be named intra and web respectively. The upper /24 space shall be subnetted as a /24 and shall be named dmz. Hosts that are to only be accessible to the institution’s intranet shall be attached to the intra subnet. Hosts that are to be publicly accessible shall be attached to the web subnet. Finally, the dmz subnet shall by way of the NLB or WAF provide NLB services and/or WEB application firewalling with deep packet inspection. NAT translations are used between the NLB or WAF and the respective TomCat application instances. |

# Program BackupBlockVolumes.ps1

**Synopsis:**

This program replicates all block volumes for the specified compartment $CompartmentName from the default region specified in $HOME/.oci/config to the specified region $TargetRegion. The default region should be the primary region of the tenant.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | Name of the compartment where the source block volumes are to be replicated from |
| $TargetRegion | string | The target region to replicate the block volume backups to. The tenant must be subscribed to the target region prior to attempting replication. |

**Examples:**

BackupBlockVolumes.ps1 “test” “us-phoenix-1”

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Zero value returned to shell plus program output to stdout and stderr |
| Failure | 1 | The specified compartment is not found, or no backup objects are found within the specified compartment |

# Program BackupBootVolumes.ps1

**Synopsis:**

This program replicates all boot volumes for the specified compartment $CompartmentName from the default region specified in $HOME/.oci/config to the specified region $TargetRegion. The default region should be the primary region of the tenant.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | Name of the compartment where the source boot volumes are to be replicated from |
| $TargetRegion | string | The target region to replicate the boot volume backups to. The tenant must be subscribed to the target region prior to attempting replication. |

**Examples:**

BackupBootVolumes.ps1 “test” “us-phoenix-1”

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Zero value returned to shell plus program output to stdout and stderr |
| Failure | 1 | The specified compartment is not found, or no backup objects are found within the specified compartment |

# Program GetActiveChildCompartment.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified active child compartment of the parent compartment ParentCompartmentName specified in tenant.json. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the active child compartment to search for |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetActiveChildCompartment.ps1 “test” “ALL”

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetActiveParentCompartment.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified active parent compartment $CompartmentName of the root compartment TenantId specified in tenant.json. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $BackupPolicy | string | The name of the policy to search for |
| $options | string |  |

**Examples:**

GetActiveParentCompartment.ps1 “VPN”

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns zero to the shell and the OCID of the parent compartment as a string value |
| Failure | 0 | Returns zero to the shell and a null value to stdout |

# Program GetBackupPolicy.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified backup policy within $CompartmentName child compartment of the parent compartment ParentCompartmentName specified in tenant.json. The program returns the OCID of the specified parent compartment if found.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $BackupPolicyName | string | The name of the backup policy to search for |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetBackupPolicy.ps1 Backup mystatejsubp01 ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetDbNode.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified database service node in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $DbSystemName | string | The database compartment name associated with the DBaaS service |
| $DbNodeName | string | The name of the DBaaS service node associated with the DBaaS service |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetDbNode.ps1 database TESTCDB myudbt01 ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetLpg.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified local peering gateway in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json that is associated with the VCN $VcnName. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VcnName | string | The name of the VCN that the LPG is associated with |
| $LpgName | string | The name of the local peering gateway |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetLpg.ps1 database databasevcn DatabaseToVpnLPG ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns zero to the shell and the OCIReturns a zero value on success along with the selected data in hash table dictionary formD of the parent compartment as a string value |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetRouterTable.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified router table in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json that is associated with the VCN $VcnName. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VcnName | string | The name of the VCN that the LPG is associated with |
| $RouterTableName | string | The name of the router table |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetRouterTable.ps1 database database\_vcn DatabaseLpgRouteTable ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetSubnet.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified subnet in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json that is associated with the VCN $VcnName. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VcnName | string | The name of the VCN that the LPG is associated with |
| $SubnetName | string | The name of the subnet |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetSubnet.ps1 web web\_vcn intra\_subnet ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetVcn.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified VCN in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VcnName | string | The name of the VCN that the LPG is associated with |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetVcn.ps1 database database\_vcn ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetVm.ps1

**Synopsis:**

This program queries the OCI cloud REST service for the specified virtual machine instance $VmName in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VmName | string | The name of the VCN that the LPG is associated with |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetVm.ps1 database myurmanp01 ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetVmBlockVol.ps1

**Synopsis:**

This program queries the OCI cloud REST service for all block volumes attached to the specified virtual machine instance $VmName in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VmName | string | The name of the VCN that the LPG is associated with |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetVmBlockVol.ps1 database myurmanp01 ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetVmBlockVolBkup.ps1

**Synopsis:**

This program queries the OCI cloud REST service for all block volume backups associated with the specified virtual machine instance $VmName in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json within the region $Region. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VmName | string | The name of the VCN that the LPG is associated with |
| $Region | string | The specified region where the volume backups are stored |

**Examples:**

GetVmBlockVolBkup.ps1 database myurmanp01 "US-PHOENIX-1"

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the backup report to stdout |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetVmBootVol.ps1

**Synopsis:**

This program queries the OCI cloud REST service for all the boot volume attached to the specified virtual machine instance $VmName in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VmName | string | The name of the VCN that the LPG is associated with |
| $options | string | Valid options are: ALL - All properties describing this object, COMPARTMENT - The parent compartment’s OCID of which this object is a member of, DISPLAYNAME - The display name of the object, OCID - The OCID of the object |

**Examples:**

GetVmBootVol.ps1 database myurmanp01 ALL

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the selected data in hash table dictionary form |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program GetVmBootVolBkup.ps1

**Synopsis:**

This program queries the OCI cloud REST service for all boot volume backups associated with the specified virtual machine instance $VmName in the child compartment $CompartmentName of ParentCompartmentName

specified in tenant.json within the region $Region. The program returns values based on valid program inputs.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VmName | string | The name of the VCN that the LPG is associated with |
| $Region | string | The specified region where the volume backups are stored |

**Examples:**

GetVmBootVolBkup.ps1 database myurmanp01 "US-PHOENIX-1"

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the backup report to stdout |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |

# Program StartVMs.ps1

**Synopsis:**

This program calls “ansible-playbook” and specifies the playbook 100\_StartVMs.yaml to start all VMs specified within 100\_StartVMs.yaml. All returned values and output are from ansible-playbook.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| NONE | NONE | NONE |

**Examples:**

StartVMs.ps1

# Program StopVMs.ps1

**Synopsis:**

This program calls “ansible-playbook” and specifies the playbook 100\_StopVMs.yaml to start all VMs specified within 100\_SopVMs.yaml. All returned values and output are from ansible-playbook.

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| NONE | NONE | NONE |

**Examples:**

StopVMs.ps1

# Program TestRestoreVmFromBkup.ps1

**Synopsis:**

This program submits requests the OCI cloud REST service for performing a restore operation test of the instance $VmName within the compartment $CompartmentName to the specified region $Region. It names the target restore instance name $NewVmName. The program gets the VM’s boot volume and builds a list of all backups for the boot volume. It then creates a boot volume with the name $NewVmName+’Boot-Vol’ to the target region $Region from the most recent incremental+full backup set, and subsequently launches an instance that attaches to the newly created boot volume. The program then reports the status of a start/stop sequence of operations. It then removes the instance and boot volume just restored, and finalizes by reporting the successful removal of the test restored objects

**Parameters:**

| Parameter | Value | Comments |
| --- | --- | --- |
| $CompartmentName | string | The name of the compartment to search for the specified object |
| $VmName | string | The name of the source VM to restore |
| $Region | string | The target region to restore to |
| $NewVmName | string | The name of the test VM to restore to |

**Examples:**

TestRestoreVmFromBkup.ps1 test mesadmt01 "US-ASHBURN-1" trmyuadmt01

**Return Values:**

| Result | Returned | Description |
| --- | --- | --- |
| Success | 0 | Returns a zero value on success along with the test restore report to stdout |
| Failure | 1 | Returns a non-zero value with stderr and stdout output |