DKC Cloud Code Base R1.0
Table of Contents

Sep 3, 2020

Copywriter Statement

IMPORTANT - THIS IS A LEGAL AGREEMENT BETWEEN YOU, ON BEHALF OF YOUR ORGANIZATION ("LICENSEE") AND DAVID KENT CONSULTING, INC, DAVID KENT CLOUD SOLUTIONS, INC, OR ITS SUBSIDIARIES (DKC). BY USING THIS SOFTWARE, OR, BY DOWNLOADING, INSTALLING, RECEIVING OR USING A PRODUCT OR SERVICE, YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS, DO NOT DOWNLOAD, INSTALL, RECEIVE OR USE THE PRODUCT OR SERVICE AND YOU WILL NOT HAVE A RIGHT TO USE ANY PART OF A PRODUCT OR SERVICE. ON OR AFTER THE DATE YOU ACCEPT THESE RESTRICTED USE LICENSE TERMS ("TERMS"), YOUR RIGHTS, OBLIGATIONS AND USE OF PRODUCT OR SERVICE PROVIDED BY DKC ARE SUBJECT TO THESE TERMS UNLESS YOU HAVE EXECUTED TERMS TO PURCHASE A SPECIFIC PRODUCT OR SERVICE FROM DKC. "PRODUCT" OR "SERVICE" INCLUDES ALL RELATED USER MANUALS AND TRAINING MATERIALS PROVIDED TO YOU ELECTRONICALLY OR PHYSICALLY ("DOCUMENTATION"), AND REVISIONS AND UPDATES THAT REPLACE OR SUPPLEMENT A PRODUCT OR SERVICE.

I. LICENSE/RIGHT TO USE

A. "Product" means DKC proprietary software products.

B. DKC Grant. DKC grants to Licensee a revocable, royalty-free, non-exclusive, non-transferable right to use Product or SaaS ("License") for Restricted Use according to the Documentation, these Terms, and any specified Measured License, for its own internal business operations at the Licensee site to which it is first provided. "Measured License" means the time period, number and type of Users, transactions, copies, seats, instructions per second (MIPS) or other quantity or measure for Product or SaaS.

C. Licensee Grant. Licensee grants DKC a royalty-free, worldwide, perpetual license to use and incorporate into DKC products any suggestion, idea, enhancement, feedback, recommendation or other information Licensee provides to DKC relating to Restricted Use Product or SaaS.

II. DELIVERY

"Delivery" means DKC has provided, and Licensee is deemed to have accepted, Product or SaaS through electronic access. DKC may provide Product for Licensee to download. Product or SaaS risk of loss passes to Licensee upon Delivery. Licensee is responsible for customs documents, clearance, and paying any customs duties, taxes and import fees.

III. LICENSEE OBLIGATIONS

1) use Product and SaaS according to these Terms, be responsible for its activity in using SaaS, and ensure its end users comply with these Terms;

2) be responsible for the accuracy, quality, legality, integrity, ownership, protection, adequate storage, security, privacy and acquisition of Data;

3) prevent unauthorized access to Product and SaaS;

4) comply with laws and regulations in using SaaS, including: data privacy, technical or personal data transmission, and anti-spam;

5) notify DKC immediately of unauthorized access to Product or SaaS, or known or suspected breach of SaaS security, and use reasonable efforts to immediately stop unauthorized access, known or suspected violation of law, or these Terms;

6) properly configure, program and operate Licensee's hardware, software, websites, Data, content, and internet to access and use SaaS;

7) comply with DKC's protocols and requirements for SaaS;

8) define and perform tests to: determine suitability of Beta Product or SaaS for Licensee's business and environment; detect defects, errors and inadequacies; recommend modifications, improvements, corrections; consult with DKC to review and analyze representative data; and use reasonable efforts to provide written test and evaluation information to DKC for Beta Product or SaaS.

B. Licensee will not:

1) permit any party to access or use Product or SaaS other than its employees; sell, resell, rent, lease or sublicense Product or SaaS, or use Product or SaaS to provide time share or other services to third parties;

2) create derivative works based on Product or SaaS, copy, frame or mirror any part of SaaS, other than copying or framing Licensee Data, content or intranet for Licensee's internal business operations;

3) decompile, disassemble, reverse engineer, access or use SaaS or Product to build a competitive product or service, copy features, functions or graphics;

4) use SaaS to store or transmit infringing, libelous, or unlawful or tortious material or transmit material in violation of privacy rights;

5) use SaaS to store or transmit viruses, worms, time bombs, Trojan horses or other harmful or malicious code, files, scripts, agents or programs; interfere with or disrupt the integrity or performance of SaaS or attempt to gain unauthorized access to SaaS or related systems;

6) transfer or use Product or SaaS at a site other than that to which it is first provided;

7) use Product or SaaS in Licensee's production environment; or infringe or misappropriate DKC's or its licensors' Intellectual Property.

C. Verification. Licensee will provide information and certification to DKC upon request to verify Licensee's Product or SaaS use according to the Agreement. Licensee will immediately pay DKC current fees for Licensee's Product or SaaS use that exceeds the Restricted Use License. THE LIMITATION OF LIABILITY SET FORTH IN SECTION IX DOES NOT APPLY TO LICENSEE'S USE EXCEEDING ITS LICENSE, OR INFRINGEMENT OR MISAPPROPRIATION OF DKC INTELLECTUAL PROPERTY OR CONFIDENTIAL INFORMATION.

IV. TERM AND TERMINATION

A. DKC may terminate this Agreement or a License at any time without notice.

B. Upon termination: 1) Licensee will immediately return or destroy Product and Documentation, cease using SaaS, and provide written certification of destruction upon request, and 2) DKC may remove or discard Licensee Data when using DKC's SaaS.

V. INTELLECTUAL PROPERTY OWNERSHIP

“Intellectual Property” means all intellectual property including, without limitation, inventions, patents, copyrights, trademarks, service marks, trade names, trade secrets, know-how, moral rights, licenses, and any other intangible proprietary or property rights, whether or not patentable or otherwise subject to legally enforceable restrictions or protections against unauthorized third party usage or whether arising by statute or common law. DKC, its affiliates or licensors own and retain all right, title and interest to and in all intellectual property in Product and SaaS, including without limitation, inventions, patents, copyrights, trademarks, service marks, trade names, trade secrets, know-how, moral rights, licenses, developments, research data, designs, layout, models, formulae, documents, drawings, plans, specifications and other DKC information, proprietary materials and all derivative works, and any other intangible proprietary or property rights, whether or not patentable or otherwise subject to legally enforceable restrictions or protections against unauthorized third party use, or whether arising by statute or common law. To the extent that any right, title or interest in or to any DKC Intellectual Property may not automatically vest in DKC by operation of law, Licensee irrevocably transfers, assigns and conveys all right, title, and interest therein to DKC. At DKC’s request and expense Licensee will promptly take any action and execute any documents necessary to vest full title in DKC or its licensor.

VI. WARRANTY DISCLAIMER

A. NOTWITHSTANDING ANY OTHER PROVISION, DKC PROVIDES RESTRICTED USE PRODUCT AND SAAS "AS IS", “AS-AVAILABLE”, WITHOUT WARRANTY, REPRESENTATION OR INDEMNITY OF ANY KIND, EXPRESS, IMPLIED, ORAL, WRITTEN, STATUTORY, OR OTHERWISE, INCLUDING WITHOUT LIMITATION, WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PURPOSE, OR NON-INFRINGEMENT, OR THAT: 1) PRODUCT OR SAAS WILL BE UNINTERRUPTED, ERROR FREE, OR OPERATE IN COMBINATION WITH ANY HARDWARE, SOFTWARE, SYSTEM, OR DATA; 2) DATA WILL BE ACCURATE OR RELIABLE; 3) THE QUALITY OF PRODUCT, SAAS, OR RESULTS OR INFORMATION OBTAINED WILL MEET LICENSEE’S REQUIREMENTS OR EXPECTATIONS; 4) ERRORS OR DEFECTS WILL BE CORRECTED; OR 5) THE SERVICE OR SERVER USED TO PROVIDE SAAS ARE FREE OF VIRUSES OR OTHER HARMFUL COMPONENTS.

B. SAAS MAY BE SUBJECT TO LIMITATION, DELAY, AND OTHER PROBLEMS INHERENT IN USE OF THE INTERNET AND ELECTRONIC COMMUNICATION AND DKC DISCLAIMS LIABILITY FOR RESULTING DELAY OR DELIVERY FAILURE.

C. LICENSEE HAS SOLE RESPONSIBILITY FOR ITS USE OF PRODUCT OR SAAS OR FULFILLING LICENSEE’S REQUIREMENTS OR OBJECTIVES IN USING RESTRICTED USE PRODUCT OR SAAS AND DKC DISCLAIMS LIABILITY FOR ANY PARTY’S ACT OR OMISSION IN USE OF OR RELIANCE UPON PRODUCT OR SAAS. LICENSEE’S SOLE REMEDY AND RECOURSE IS TO STOP USING PRODUCT OR SAAS AND SECTION V(B) WILL APPLY. THIS SECTION VIII DESCRIBES DKC’S SOLE LIABILITY FOR PRODUCT OR SAAS WARRANTY.

VII. LIMITATION OF LIABILITY

A. DKC IS NOT LIABLE FOR (i) ANY CONSEQUENTIAL, INDIRECT, SPECIAL, PUNITIVE OR INCIDENTAL DAMAGES, (ii) ANY INTERRUPTION OF SERVICE, BUSINESS OR OPERATIONS, COST OF COVER, GOODWILL, LOST DATA, PROFITS OR REVENUE, OR FAILURE OF A REMEDY TO ACHIEVE ITS ESSENTIAL PURPOSE. DKC IS NOT LIABLE FOR LOSS OR DAMAGES, DIRECT OR INDIRECT, RESULTING FROM LICENSEE’S USE OR ATTEMPTED USE OF PRODUCT OR SAAS, INCLUDING, WITHOUT LIMITATION, CLAIMS BASED UPON THEIR SUFFICIENCY, RELIABILITY, TIMELINESS, QUALITY, SECURITY, SUITABILITY, TRUTH, AVAILABILITY, COMPLETENESS OR ACCURACY.

B. EXCEPT FOR DEATH, BODILY INJURY, WILLFUL MISCONDUCT OR FRAUD, AND SOLELY TO THE EXTENT PROXIMATELY CAUSED BY DKC, DKC IS NOT LIABLE FOR ANY DAMAGES THAT EXCEED THE AMOUNT DKC RECEIVED FROM LICENSEE FOR THE RESTRICTED USE PRODUCT OR SAAS GIVING RISE TO A CLAIM. THIS SECTION VII DESCRIBES THE TOTAL EXTENT OF DKC’S LIABILITY FOR PRODUCT OR SAAS.

VIII. COMPLIANCE WITH LAWS; EXPORT

Licensee must comply with U.S., foreign, and international laws and regulations, including without limitation the U.S. Export Administration and Treasury Department's Office of Foreign Assets Control regulations, and other anti-boycott and import regulations. Licensee agrees: (i) that the export, re-export, transfer, re-transfer, sale, supply, access to, or use of Product or SaaS to or in a country other than the country in which the Product or SaaS was first

provided to Licensee, or to, by, or for a different end user or end use, may require a U.S. or other government license or other authorization; and (ii) not to, directly or indirectly, export, re-export, transfer, re-transfer, sell, supply, or allow access to or use of Product or SaaS to, in, by, or for sanctioned, embargoed, or prohibited countries, persons, or end uses under U.S. or other applicable law (collectively, “Prohibited Use“). Licensee is responsible for screening for Prohibited Use and obtaining any required licenses or other authorizations and shall indemnify DKC for any violation by Licensee of any applicable export controls and/or economic sanctions laws and regulations. DKC may terminate the Agreement and License immediately if DKC determines, in its sole discretion, that Licensee has breached, intends to breach, or insists upon breaching any of the provisions in this clause.

IX. U.S. GOVERNMENT USERS

Product and SaaS includes “Commercial Computer Software” and “Commercial Computer Software Documentation.” In accordance with Section 12.212 of the Federal Acquisition Regulations (FAR) and Sections 227.7202-1 through 227.7202-4 of the Defense Federal Acquisition Regulation Supplement (DFARS), any use, duplication or disclosure of Product or SaaS by the U.S. Government or any of its agencies will be governed by and subject to all of the Terms, conditions, restrictions, and limitations of the Agreement. Use of Product or SaaS is agreement by the U.S. Government that Product or SaaS includes "commercial computer software" and "commercial computer software documentation" and is acceptance of the rights and restrictions in the Agreement. If for any reason Product or SaaS is not considered commercial or the Agreement Terms are otherwise deemed not to apply, the Product or SaaS will be deemed to be provided with “Restricted Rights” as defined in FAR 52.227-14(a) and FAR 52.227-14(g)(4) (Alt III), or DFARS 252.227-7014(a)(15) and DFARS 252.227-7014(b)(3), as applicable.

X. MISCELLANEOUS

A. Integration. The Agreement sets forth the entire agreement of the parties and supersedes all prior oral and written agreements and understandings with respect to its subject matter. No waiver or modification of any provision of the Agreement is binding unless in writing signed by both parties. DKC’s failure to exercise a right or remedy is not a waiver. If an Agreement term is invalid, illegal or unenforceable, it will not affect any other Agreement term. The Agreement may not be assigned by Licensee.

B. Law. The laws of Michigan, USA, govern the Agreement, excluding conflict of law principles which would require application of the law of any other jurisdiction. Each party waives the right to jury trial for any legal action. The United Nations Convention on Contracts for the International Sale of Goods and the Uniform Computer Information Transaction Act as adopted do not apply.

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-accessible	null	
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	

Key	Returned	Description
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
myRegion	string	The datacenter region where the object was created

Examples:

```
GetBackupPolicies $myCompartment $myRegion
```

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	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 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
myRegion	string	The datacenter region where the block volume was created

Examples:

```
GetBlockVolumes $myVM $myRegion
```

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

Key	Returned	Description
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 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
myRegion	string	The datacenter location where the boot volume was created

Examples:

```
GetBootVolumes $myVM $myRegion
```

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 GetCluster

Synopsis:

This function checks for and returns properties of the selected Kubernetes cluster

Parameters:

Parameter	Value	Comments
$myCompartment	string	OCID of the compartment the cluster is a member of
$myClusterName	string	The name of the Kubernetes cluster to search for
$myRegion	string	The data center region where the cluster service was created in

Examples:

```
GetCluster $myCompartment $myClusterName “US-ASHBURN-1”
```

Return Values:

Key	Returned	Description
kubernetes-version	string	The version of the Kubernetes cluster
id	string	The OCID of the object
lifecycle-state	string	The life cycle state of the object, should always be ACTIVE
compartment-id	string	The OCID of the compartment the object is a member of
options	array	Cluster settings that were defined at time of creation or subsequently modified
options.admission-controller-options	array	Security options of the cluster
options.add-ons	array	Service feature add-ons for the cluster
options.add-ons.is-kubernetes-dashboard-enabled	boolean	Dashboard enabled if true
options.add-ons.is-tiller-enabled	boolean	Tiller is enabled if true

Key	Returned	Description
available-kubernetes-upgrades	unused	
metadata	array	Detail regarding the creation metadata regarding the cluster, such as the date the cluster was created
endpoints	array	Public end points of the cluster, more info needed
vcn-id	string	The OCID of the VCN the cluster is bound to
name	string	The name of the cluster

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
myRegion	string	The datacenter region where the node was created

Examples:

```
GetDbNodeName $myDbSystems $myDbNodeName $myRegion
```

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
myRegion	string	The data center region where the object was created in

Examples:

```
GetDbSystems $myCompartment $myRegion
```

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	string	The OCID of the DBaaS resource
data.instance-id	string	Same as 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 GetDRGs

Synopsis:

This function checks for and returns properties of all dynamic routing gateways within a compartment

Parameters:

Parameter	Value	Comments
myVcn	array	The dictionary object of the compartment as returned by the function SelectVcn
myRegion	string	The data center region where the object was created in

Examples:

```
GetDRGs $myVcn $myRegion
```

Return Values:

Key	Returned	Description
data	array	Array of objects that describe all DRGs found within the compartment
compartment-id	string	The OCID of the compartment where the DRG resides
display-name	string	The name of the DRG
id	string	The OCID of the DRG
lifecycle-state	string	Must be AVAILABLE
time-created	date	The date that the DRG was created

Function GetFileSystems

Synopsis:

This function checks for and returns properties of all file systems within a compartment

Parameters:

Parameter	Value	Comments
myCompartment	array	The dictionary object of the compartment as returned by the function GetActiveChildCompartment
myAvailabilityDomain	string	The name of the availability domain where the file system was originally created
myRegion	string	The datacenter region where the file system was created

Examples:

```
GetFileSystems $myCompartment $myAvailabilityDomain $myRegion
```

Return Values:

Key	Returned	Description
data	array	Array of objects that describe all File Systems found within the compartment

Function GetFileSystem

Synopsis:

This function checks for and returns properties of the selected file system

Parameters:

Parameter	Value	Comments
myFileSystems	array	The dictionary object of the file systems as returned by the function GetFileSystems
myFileSystemName	string	The name of the file system to search for

Examples:

```
GetFileSystem $myFileSystems $myFileSystemName
```

Return Values:

Key	Returned	Description
display-name	string	The name of the file system
time-created	date	The date stamp when the file system was created
id	string	The OCID of the object
life-cycle-state	string	The state of the object, should always be ACTIVE
compartment-id	string	The OCID of the compartment the file system is a member of
metered-bytes	int	The number of bytes metered during the billing period
kms-key-id	string	unused
availability-domain	string	The availability domain where the object resides

Function GetIGWs

Synopsis:

This function checks for and returns a list of Internet Gateways found within a compartment.

Parameters:

Parameter	Value	Comments
$myVCN	array	The dictionary object of the VCN that holds the Internet Gateways
$myRegion	string	The data center region to search in

Examples:

```
GetIGWs $myVCN $myRegion
```

Return Values:

Key	Returned	Description
lifecycle-state	string	The lifecycle state of the object, must always be AVAILABLE
vcn-id	string	The OCID of the VCN that holds the NAT Gateway
is-enabled	boolean	True or False
time-created	date	The date stamp when the object was created
id	string	The OCID of the Internet Gateway
compartment-id	string	The OCID of the compartment housing the objects
display-name	string	The name of the Internet Gateway

Function GetIpSecTunnels

Synopsis:

This function checks for and returns all IP Sec tunnel objects that are associated with $myIpSecConnection

Parameters:

Parameter	Value	Comments
$myIpSecConnection	array	The dictionary object of the VCN that holds the Internet Gateways

Examples:

```
GetIpSecTunnels
```

Return Values:

Key	Returned	Description
compartment-id	string	The OCID where the object was created
cpe-id	string	The OCID of the CPE record
cpe-local-identifier	string	The public IP address of the IP-Sec Connection
cpe-local-identifier-type	string	Should be IP_ADDRESS
display-name	string	The name of the IP-Sec Connection
drg-id	string	The name of the DRG associated with this IP-Sec Connection
id	string	The OCID of this IP-Sec Connection
lifecycle-state	string	The lifecycle state of this object
time-created	date	The date stamp when this object was created

Function GetIpSecConnections

Synopsis:

This function checks for and returns a list of IP-Sec Connections found within a compartment.

Parameters:

Parameter	Value	Comments
$myVCN	array	The dictionary object of the VCN that holds the Internet Gateways
$myRegion	string	The data center region to search in

Examples:

```
GetIpSecConnections $myVCN $myRegion
```

Return Values:

Key	Returned	Description
compartment-id	string	The OCID where the object was created
cpe-id	string	The OCID of the CPE record
cpe-local-identifier	string	The public IP address of the IP-Sec Connection
cpe-local-identifier-type	string	Should be IP_ADDRESS
display-name	string	The name of the IP-Sec Connection
drg-id	string	The name of the DRG associated with this IP-Sec Connection
id	string	The OCID of this IP-Sec Connection
lifecycle-state	string	The lifecycle state of this object
time-created	date	The date stamp when this object was created

Function GetKbClusters

Synopsis:

This function checks for and returns a list of Kubernetes clusters found within a compartment.

Parameters:

Parameter	Value	Comments
$myCompartment	array	The dictionary object of the compartment as returned by GetActiveChildCompartment
$myRegion	string	The data center region to search in

Examples:

```
GetKbCluster $myCompartment $myRegion
```

Return Values:

Key	Returned	Description
region	string	The data center region where the cluster service is located
compartment_name	string	The name of the compartment where the cluster is a member of
kbcluster_name	string	The name of the cluster
metadata	array	A list of objects describing the cluster creation, removal, and update status
lifecycle-state	string	The lifecycle state, should be AVAILABLE
kubernetes-version	string	The version of Kubernetes deployed to this cluster
available-kubernetes-upgrades	string	Version upgrades available to the cluster
name	string	The name of the cluster
compartment-id	string	The OCID of the compartment holding the cluster
vcn-id	string	The OCID of the VCN the cluster is bound to
options	array	A list of KB options deployed on the cluster service

Key	Returned	Description
endpoints	array	Should always be “kubernetes”
id	string	The OCID of the cluster

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 GetMountTargets

Synopsis:

This function checks for and returns properties of all mount targets that are members of a compartment within the availability domain $myAvaiulabilityDomain. It returns a dictionary object containing the properties of the mount targets.

Parameters:

Parameter	Value	Comments
myCompartment	array	The dictionary object of the compartment
$myAvailabilityDomain	string	The name of the availability domain to search for mount targets
$myRegion	string	The datacenter region where the mount target was created

Examples:

```
GetMountTargets $myCompartment $myAvailabilityDomain $myRegion
```

Return Values:

Key	Returned	Description
data	array	Array of objects that describe all mount targets

Function GetMountTarget

Synopsis:

This function checks for and returns the mount target object from $myMountTargets.

Parameters:

Parameter	Value	Comments
myMountTargets	array	The dictionary object of all mount targets as returned by the function GetMountTargets
$myMountTargetName	string	The name of the mount target to search for

Examples:

```
GetMountTarget $myMountTargets $myMountTargetName
```

Return Values:

Key	Returned	Description
subnet-id	string	The OCID of the subnet the mount target is bound to
time-created	date	The date stamp the object was created on
id	string	The OCID of the object
export-set-id	string	An alias to *id* with the second field replacing the field “mounttarget” with “exportset”.
lifecycle-state	string	The life cycle status of the object, should always be “ACTIVE”
private-ip-ids	array	The OCID of any private IPs bound to the mount target’s network interface
compartment-id	string	The compartment OCID the object is a member of
availability-domain	string	The name of the availability domain where the file system object was created in
nsg-ids	array	The network security groups applied to this object

Function GetNatGWs

Synopsis:

This function checks for and returns a list of NAT Gateways found within a compartment.

Parameters:

Parameter	Value	Comments
$myVCN	array	The dictionary object of the VCN that holds the NAT Gateways
$myRegion	string	The data center region to search in

Examples:

```
GetNatGWs $myVCN $myRegion
```

Return Values:

Key	Returned	Description
lifecycle-state	string	The lifecycle state of the object, must always be AVAILABLE
nat-ip	string	The public IP assigned to the NAT Gateway
vcn-id	string	The OCID of the VCN that holds the NAT Gateway
time-created	date	The date stamp when the object was created
id	string	The OCID of the NAT Gateway
block-traffic	boolean	Usually $False
compartment-id	string	The OCID of the compartment housing the objects
display-name	string	The name of the NAT Gateway

Function GetNodePool

Synopsis:

This function checks for and returns the node pool object that is associated with $myCluster within the compartment $myCompartment.

Parameters:

Parameter	Value	Comments
$myCluster	array	The dictionary object of the cluster from which to search for node pools
$myNodePoolName	string	The name of the node pool to search search for
$myRegion	string	The datacenter region where the node pool was created

Examples:

```
GetNodePool $myCluster $myNodePoolName $myRegion
```

Return Values:

Key	Returned	Description
node_shape	string	The shape that has been applied to the nodepool
quantity_per_subnet	int	Optional, if used, defines the numder of nodes per subnet
cluster-id	string	The OCID of the Kubernetes cluster the node pool is attached to
ssh-public-key	string	The SSH key on the nodes at the time of build
node-image-ocid	string	The OCID of the source image used to build to nodes
subnet-ids	array	The OCIDs of the subnets the nodes are associated with. Should only be one subnet OCID when subnets are built to span all availability domains within a region. This is the best practice and is also the API default.
name	string	The name of the node pool
node-image-name	string	The name of the image used to build the nodes in the pool

Key	Returned	Description
compartment-id	string	The OCID of the compartment the node pool is a member of
node-source	array	Contains more detail regarding the source used to build the nodes in the pool
kubernetes-version	string	The version of Kubernetes pushed to the nodes within the pool. This value must be the same as the Kubernetes cluster the node pool is attached to. The DKC codebase enforces this when used to manage the cluster.
node-config-details	array	Details regarding the node pool configureation
node-config-details.placement-configs	array	Details regarding the availability domains and subnets to which nodes are installed into. The DKC code will by default disburse nodes across all three availability domains for each node pool. In doing so, the code enforces creation of a fully resilient node pool.. The list shown below will always hold 6 elements, 3 elements for the 3 availability domains, and 3 elements for the subnet. The subnet OCID will be consistently the same across all elements. This means all nodes within a pool are only associated with a single subnet but are evenly dispersed in groups of 3 across all availability domains.
node-config-details.placement-configs.availability-domain	string	One entry for each availability domain within the region in which the cluster and node pool are created
node-config-details.placement-configs.subnet-id	string	The OCID to which the nodes in the pool are assigned.

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 GetNetSecurityGroup

Synopsis:

This function checks for and returns properties of the selected network security group

Parameters:

Parameter	Value	Comments
$myCompartment	array	The compartment dictionary object as returned by GetActiveChildCompartment
$myNetworkSecurityGroup	string	The name of the security group to search for
$myRegion	string	The name of the datacenter region where the security group was created

Examples:

```
GetNetSecurityGroup $myCompartment $myNetworkSecurityGroup $myRegion
```

Return Values:

Key	Returned	Description
display-name	string	The name of the security group
id	string	The OCID of the object
vcn-id	string	The VCN the security group is associated with
compartment-id	string	The OCID of the compartment the object is a member of
time-created	date	The date the object was created
lifecycle-state	string	The state of the object, should always be AVAILABLE

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
$myCompartment	array	The dictionary object of the compartment
$myVcn	array	The dictionary object of the VCN that holds the LPGs
$myRegion	string	The region where the VCN and subnet were created

Examples:

```
GetSubnet $myCompartment $myVCN $myRegion
```

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

Key	Returned	Description
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
myRegion	string	Optional, specify the region where the VCN object was created. The default region defined in ~.oci/config is used in the absence of specifying a region.

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

Key	Returned	Description
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-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-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

Key	Returned	Description
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
myRegion	String	Region where the VM is located at

Examples:

```
GetVMs $myCompartment $myRegion
```

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 GetVnic

Synopsis:

This function checks for and returns properties regarding the virtual network interface within the specified compartment

Parameters:

Parameter	Value	Comments
$myVm	array	A dictionary object that describes the VM as returned by GetVm
myRegion	string	Optional, specify the region where the VCN object was created. The default region defined in ~.oci/config is used in the absence of specifying a region.

Examples:

```
GetVCN $myVm $myRegion
```

Return Values: (in JSON)

Key	Returned	Description
subnet-id	string	The OCID of the subnet the VNIC is associated with
availability-domain	string	The availability domain where the VNIC is located
id	string	The OCID of the VNIC attachment
compartment-id	string	The OCID of the compartment the object is a member of
vlan-tag	int	The VLAN number the NIC is assigned to
nic-index	int	The VNIC number attached to the VM
time-created	date	The date the object was created
lifecycle-state	string	The lifecycle state of the object, should always be ‘AVAILABLE’
vnic-id	string	The OCID of this object
display-name		Not used
instance-id	string	The OCID of the instance the VNIC is attached to

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
myRegion	string	The datacenter region where the resource was created

Examples:

```
GetVmNicAttachment $myVM “US-ASHBURN-1”
```

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
myRegion	string	The datacenter region where the backups are stored

Examples:

```
GetVmBootVolBackups $myCompartment $myRegion
```

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

Key	Returned	Description
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

Key	Returned	Description
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 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

Key	Returned	Description
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

Key	Returned	Description
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 SelectDbSystem

Synopsis:

This function returns the dictionary object from $myIGWs that matches the string value of $myIgwName

Parameters:

Parameter	Value	Comments
$myDbSystem	string	Name of the DB system to search for
$myDbSystems	array	Dictionary object describing all DB Systems within a compartment

Examples:

```
SelectDbSystem $myDbSystem $MyDbSystems
```

Return Values:

Key	Returned	Description
data-storage-percentage	int	Percentage of storage allocated to the database from the storage aggregate
ssh-public-keys	array	The SSH public keys that are keyed with the service node(s)
db-system-options	object	The class of storage, may be ASM or LVM
fault-domains	array	The list of fault domains the service node and appliance are associated with
cluster-name	string	If implemented as a RAC cluster, the name of the cluster
subnet-id	string	The OCID of the subnet the DB System is associated with
time-created	date	The date the appliance was created on
reco-storage-size-in-gb	int	The amount of usable storage in the aggregate
database-edition	string	The licensed edition of Oracle Database, may be STANDARD, ENTERPRISE, HIGH_PERFORMANCE, EXTREAME_PERFORMANCE
cpu-core-count	int	The number of CPUs licensed

Key	Returned	Description
display-name	string	The display name of the appliance
domain	string	The Oracle DNS FQND for this appliance
node-count	int	The number of nodes in the cluster
hostname	string	The name of the service node
shape	string	The VM shape applied to the service node(s)
disk-redundancy	string	The level of disk redundancy for the appliance
time-zone	string	The time zone applied to the appliance
availability-domain	string	The name of the availability domain where the appliance was created in
Listener port	int	The TCP port the appliance listens on
id	string	The OCID of the appliance
lifecycle-state	string	The availability state of the appliance, should always be AVAILABLE
data-storage-size-in-gb	int	The size of the raw storage aggregate supporting the database
version	string	The version of *grid control* supporting this appliance
scan-dns-record-id	string	The OCID of t he appliance's DNS record

Function SelectDRG

Synopsis:

This function returns the dictionary object from $myDRGs that matches the string value of $myDRGName

Parameters:

Parameter	Value	Comments
$myDRGs	array	Dictionary object describing all DRGs within a compartment as returned by GetDRGs
$myDrgName	string	The name of the DRG to search for

Examples:

```
SelectDRG $myDRGs $myDrgName
```

Return Values:

The DRG object that represents the dynamic router gateway

Function SelectIGW

Synopsis:

This function returns the dictionary object from $myIGWs that matches the string value of $myIgwName

Parameters:

Parameter	Value	Comments
$myIgwName	string	Name of the IGW to search for
$myIGWs	array	Dictionary object describing all IGWs within a compartment

Examples:

```
SelectIGW $myIgwName $myIGWs
```

Return Values:

Key	Returned	Description
compartment-id	string	The OCID of the compartment the object is a member of
display-name	string	The name of the object
id	string	The OCID of the object
lifecycle-state	string	The life cycle status of the object, should be AVAILABLE for a IGW that is in use
nat-ip	string	The public IP of the NGW
time-created	date	The date stamp when the object was created
vcn-id	string	The VCN OCID that the NGW is associated with

Function SelectIpSecTunnel

Synopsis:

This function returns the dictionary object from $myIpConnections the connection name $myIpConnectionName

Parameters:

Parameter	Value	Comments
$myIpConnections	array	Dictionary object of all IP connections as returned by GetIpConnections
$myIPSecConnectionName	string	The name of the IP connection to search for

Examples:

```
SelectIpSecTunnel $myIpConnections $myIpSecConnectionName
```

Return Values:

The object is returned if found

Function SelectKbCluster

Synopsis:

This function returns the dictionary object from $myKbClusters the connection name $myKbClusterName

Parameters:

Parameter	Value	Comments
$myKbClusters	array	Dictionary object of all Kubernetes clusters as returned by GetKbClusters
$myKbClusterName	string	The name of the cluster to search for

Examples:

```
SelectKbCluster $myKbClusters $myKbClusterName
```

Return Values:

The object is returned if found

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

Key	Returned	Description
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 SelectNGW

Synopsis:

This function returns the dictionary object from $myNGWs that matches the string value of $myNgwName

Parameters:

Parameter	Value	Comments
$myNgwName	string	Name of the NGW to search for
$myNGWs	array	Dictionary object describing all NGWs within a compartment

Examples:

```
SelectNGW $myNgwName $myNGWs
```

Return Values:

Key	Returned	Description
compartment-id	string	The OCID of the compartment the object is a member of
display-name	string	The name of the object
id	string	The OCID of the object
lifecycle-state	string	The life cycle status of the object, should be AVAILABLE for a NGW that is in use
nat-ip	string	The public IP of the NGW
time-created	date	The date stamp when the object was created
vcn-id	string	The VCN OCID that the NGW is associated with

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 SelectSecurityList

Synopsis:

This function returns the security list from the dictionary object $mySecLists that matches $mySecurityListName

Parameters:

Parameter	Value	Comments
$mySecurityLists	array	Dictionary object of security lists
$mySecurityListName	string	Name of the object to search for

Examples:

```
SelectSecurityList $mySecurityListName $mySecurityLists
```

Return Values:

Key	Returned	Description
compartment-id	string	The compartment OCID where the subnet resides
display-name	string	The name of the object
egress-security-rules	array	List of defined security rules for outbound traffic
id	string	The OCID of the object
ingress-security-rules	array	List of the defined security rules for inbound traffic
lifecycle-state	date	The date stamp when the object was created
vcn-id	string	The OCID of the VCN the security list 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

Key	Returned	Description
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 $myVcnName $myVCNs
```

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

Key	Returned	Description
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.

Parameter	Value	Comments
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
$SourceRegion	string	The source region from which to replicate the block volume backups 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-ashburn-1” “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
$SourceRegion	string	The source region from which to replicate the boot volume backups 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-ashburn-1” “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 BackupReport.ps1

Synopsis:

This program reports on all VM instance backups for the compartment $CompartmentName within the region $Region.

Parameters:

Parameter	Value	Comments
$CompartmentName	string	Name of the compartment where the backup images are stored
$Region	string	The datacenter region where the backup images are stored

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 DelNodePool.ps1

Synopsis:

This program either simulates deletion or deletes the node pool $NodePoolName from Kubernetes cluster $ClusterName within compartment $CompartmentName.

Parameters:

Parameter	Value	Comments
$CompartmentName	string	Name of the compartment where the node pool is located within
$ClusterName	string	The name of the cluster to which the node pool is attached
$NodePoolName	string	The name of the node pool to remove
$Region	string	The datacenter region where the nodepool was created
$options	string	“YES” or “yes” will force delete the nodepool, any other entry or null will prompt the user to confirm prior to deletion.

Examples:

```
DelNodePool.ps1 “tst comp” “DKCUKBC01” “DKCUKBC01P01” “US-ASHBURN-1” no
```

Return Values:

Result	Returned	Description
Success	0	Zero value returned to shell plus program output to stdout and stderr
Failure	1	The specified object is not found, or an error with removing the node pool

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
$options	string	Valid options include “ALL”, “DISPLAYNAME”, “OCID”

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
$Region	string	The datacenter region where the object was created
$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
$Region	string	The data center region where the object was created in
$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 "US-ASHBURN-1" 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 GetDbSystem.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified database system 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
$Region	string	The data center region where the object was created in
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of the DB System, SHAPE - the shape applied to the DB System, EDITION - The major edition of Oracle Database licensed, such as Oracle Enterprise Edition, HOSTNAME - The host name oof the DB System's service node(s), LISTENER - The TCP port the DB System is set to listen on, NODECOUNT - The number of service nodes servicing the DB System, SSHKEYS - The ssh public keys configured for SSH key authentication to the DB System

Examples:

```
GetDbSystem.ps1 database TESTCDB "US-ASHBURN-1" 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 GetDRG.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified Dynamic Routing gateway in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json in 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
$DrgName	string	The name of the dynamic routing gateway
$Region	string	The region where the LPG was created
$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:

```
GetDRG.ps1 database databasevcn "dbs_drg" "US-ASHBURN-1" 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 GetExport.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified file system export object in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json within the data center $AvailabilityDomain. 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
$MountTargetName	string	The name of the mount target to search for
$AvailabilityDomain	string	The availability domain where the file system was created
$ExportPath	string	The path of the export
$Region	string	The datacenter region where the mount target was created
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of the object

Examples:

```
GetExport.ps1 dbs_comp DKCUFS01_MT “clNq:US-ASHBURN-AD-1” “US-ASHBURN-1” “/bin” 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 GetFileSystem.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified file system object in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json within the data center $AvailabilityDomain. 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
$FileSystemName	string	The name of the file system object to search for
$AvailabilityDomain	string	The availability domain where the file system was created
$Region	string	The datacenter region where the file system was created
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of the object

Examples:

```
GetFileSystem.ps1 dbs comp DKCUFS01 “clNq:US-ASHBURN-AD-1” “US-ASHBURN-1” 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 GetIGW.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified Internet gateway in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json that is associated with the VCN $VcnName in 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
$VcnName	string	The name of the VCN that the LPG is associated with
$IgwName	string	The name of the Internet gateway
$Region	string	The region where the LPG was created
$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:

```
GetIGW.ps1 database databasevcn “dbs_igw” “US-ASHBURN-1” 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 GetKbCluster.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified Kubernetes cluster object in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json within the data center $AvailabilityDomain. 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
$ClusterName	string	The name of the Kubernetes cluster object to search for
$Region	string	The datacenter region where the cluster service was created
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of the object

Examples:

```
GetKbCluster.ps1 web DKCUKBC01 “clNq:US-ASHBURN-AD-1” 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
$Region	string	The region where the LPG was created
$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 "US-ASHBURN-1" 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 GetNGW.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified NAT gateway in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json that is associated with the VCN $VcnName in 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
$VcnName	string	The name of the VCN that the LPG is associated with
$NgwName	string	The name of the NAT gateway
$Region	string	The region where the LPG was created
$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:

```
GetNGW.ps1 database databasevcn db ngw "US-ASHBURN-1" 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 GetNsg.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified security group in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json that is associated with the VCN $VcnName in 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
$NetworkSecurityGroup	string	The name of the security group to search for
$Region	string	The region where the LPG was created
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of the object

Examples:

```
GetNsg.ps1 dbs comp myNsg “US-ASHBURN-1” 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 GetMountTarget.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified mount target object in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json within the data center $AvailabilityDomain. 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
$MountTargetName	string	The name of the file system object to search for
$AvailabilityDomain	string	The availability domain where the file system was created
$Region	string	The datacenter region where the mount target was created
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of the object

Examples:

```
GetMountTarget.ps1 dbs_comp DKCUFS01_MT “clNq:US-ASHBURN-AD-1” “US-ASHBURN-1” 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 GetNodePool.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified node pool object in the child compartment $CompartmentName of ParentCompartmentName specified in tenant.json for the cluster $ClusterName. 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
$ClusterName	string	The name of the cluster to search for
$NodePoolName	string	The name of the node pool to search for
$Region	string	The datacenter region where the nodepool was created in
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of the object

Examples:

```
GetNodePool.ps1 “tstContainer” “DKCUKBC01” “DKCUKBC01P01”  “US-ASHBURN-1 “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 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 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
$VcnName	string	The name of the VCN that the LPG is associated with
$RouterTableName	string	The name of the router table
$Region	string	The name of the region where the route table was created
$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 "US-ASHBURN-1" 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 GetSecurityList.ps1

Synopsis:

This program queries the OCI cloud REST service for the specified security list 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
$SecurityListName	string	The name of the security list
$Region	string	The region where the object was created
$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:

```
GetSecurityList.ps1 auto_comp auto_vcn "Default Security List for auto_vcn" "US-ASHBURN-1" 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 in 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
$VcnName	string	The name of the VCN that the LPG is associated with
$SubnetName	string	The name of the subnet
$Region	string	The region where the subnet and VCN were created
$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 "US-PHOENIX-1" 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
$Region	string	Optional, specify the region where the VCN object was created. The default region defined in ~.oci/config is used in the absence of specifying a region.

Examples:

```
GetVcn.ps1 database database_vcn ALL "US-ASHBURN-1"
```

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
$Region	string	The region where the VM is located at
$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 "US-ASHBURN-1" 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 the first block 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.

WARNING! This program assumes only 1 block volume is attached per host and is based on the client EA.

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 datacenter region where the block volume was created in
$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 "US-ASHBURN-1" 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 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
$Region	string	The datacenter location where the boot volume was created
$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 "US-ASHBURN-1" 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 GetVnic.ps1

Synopsis:

This program queries the OCI cloud REST service for all the virtual network interfaces 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
$Region	string	The datacenter region where the object was created in
$options	string	Valid options are: ALL - All properties describing this object, OCID - The OCID of this object, ATTACHMENTID - The OCID of the attachment of this VNIC

Examples:

```
GetVnic web_comp DKCUVMT01 “US-ASHBURN-1” 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 LsBackupPolicy.ps1

Synopsis:

This program queries the OCI cloud REST service for all backup policies 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
$Region	string	The specified region where the DB Systems are located in
$option	string	ALL - All properties for all backup policies in the compartment, NAME - List only the names of the backup policies, SCHEDULE - List each policy and its associated backup schedules

Examples:

```
LsBackupPolicy.ps1 backup_comp “us-ashburn-1” ALL
```

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 LsDBaaS.ps1

Synopsis:

This program queries the OCI cloud REST service for all DB Systems 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
$Region	string	The specified region where the DB Systems are located in
$option	string	ALL - All data for DB Systems found in the compartment, NAME - Return just the names of DB Systems found in the compartment

Examples:

```
LsDBaaS.ps1 database "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 LsVm.ps1

Synopsis:

This program queries the OCI cloud REST service for all virtual machine instances 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
$Region	string	The specified region where the VMs are located
$option	string	ALL - All data for VMs found in the compartment, NAME - Return just the names of VMs found in the compartment

Examples:

```
LsVm.ps1 database "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 start_stop_DBaaS.ps1.ps1

Synopsis:

This program starts or stops the database system node for DB system $myDbName for the node $myNodeName in the compartment $myCompartment.

Parameters:

Parameter	Value	Comments
$myCompartment	string	The name of the compartment the object is a member of
$myDbName	string	The display name of the DB System
$myNodeName	string	The name of the node that is a member of the DB system. There may be one or two DB nodes that are members of the DB system.
$myAction	string	The action to take on the node. Valid input are either “START” or “STOP”

Examples:

```
200 start stop DBaaS.ps1 dbs comp TEST2 dkcutdb02 STOP
```

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 100_start_stop_VM.ps1

Synopsis:

This program starts or stops the specified VM $VmName within the compartment $CompartmentName within the region $Region. Passing an option of $Start will start the VM. Passing an option of $Stop will stop the VM. The stop of the VM is hard, and the start of the VM is graceful.

Parameters:

Parameter	Value	Comments
Compartment	string	The compartment where the VM is a member of
VmName	string	The name of the VM to start or stop
Region	string	The datacenter location where the VM was created in
options	string	“START” will start the VM, “STOP” will stop the VM

Examples:

```
100_start_stop_VM.ps1 dbs_comp DKCUBASP01 “US-ASHBURN-1” START
```

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 200_start_stop_DBaaS.ps1

Synopsis:

This program starts or stops the specified VM $DBaaS within the compartment $CompartmentName within the region $Region. Passing an option of $Start will start the VM. Passing an option of $Stop will stop the VM. The stop of the VM is hard, and the start of the VM is graceful.

Parameters:

Parameter	Value	Comments
Compartment	string	The compartment where the VM is a member of
DbSystemName	string	The display name or the database compartment name
DbNodeName	string	The name of the DBaaS service node
Region	string	The datacenter location where the VM was created in
options	string	“START” will start the VM, “STOP” will stop the VM

Examples:

```
100_start_stop_DBaaS.ps1 dbs_comp DKCUTEST dkcudbt01 “US-ASHBURN-1” START
```

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 200_updateDBaaS.ps1

Synopsis:

The purpose of this program is to either change a DB System's shape or to load a new set of public SSH keys for SSH key authentication. The DB System and service node(s) must be in an available state when the program is run. The private database may be open or closed when run.

This program modifies the DB system $DBaaS within the compartment $CompartmentName within the region $Region with the specified option $option with the parameter ChangeParameter.

Parameters:

Parameter	Value	Comments
Compartment	string	The compartment where the VM is a member of
DbSystemName	string	The display name or the database compartment name
option	string	Either "SSHKEY" to upload new SSH keys to the DB System, or "SHAPE" to change the service node's virtual machine size.
ChangeParameter	string	Either the directory where public SSH key files are stored, or a valid VM shape for the service node
Region	string	The datacenter location where the VM was created in

Examples:

```
200_updateDBaaS.ps1 dbs_comp DKCUTEST SSHKEY "/home/ansible/prod/ssh_keys" "us-ashburn-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 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

Playbook 001_CreateCompartments_NWMSU.yaml

Synopsis:

This ANSIBLE playbook creates compartments within the tenancy. Creates parents, then creates children

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
tenancy_id	string	The OCID of the tenancy
acad_comp	string	The name of the academic parent compartment
acad_desc	string	A description of the compartment
student_comp	string	The name of the student parent compartment
student_desc	string	A description of the compartment
parent	string	The name of the Banner tenancy parent compartment
Pdescription	string	A description of the compartment
AutomationCompartment	string	The name of the automation child compartment
AutomationCompartmentDesc	string	A description of the compartment
BackupCompartment	string	The name of the compartment that contains backup policy objects
BackupCompartmentDesc	string	A description of the compartment
BastionCompartment	string	The name of the Bastion compartment
BastionCompartmentDesc	string	A description of the compartment
DatabaseCompartment	string	The name of the database compartment
DatabaseCompartmentDesc	string	A description of the compartment
DrCompartment	string	The name of the disaster recovery compartment
DrCompartmentDesc	string	A description of the compartment
DmzCompartment	string	The name of the DMZ compartment
DmzCompartmentDesc	string	A description of the compartment

Parameter	Value	Comments
TestCompartment	string	The name of the Test compartment
TestCompartmentDesc	string	A description of the compartment
VpnCompartment	string	The name of the VPN compartment
VpnCompartmentDesc	string	A description of the compartment
WebCompartment	string	The name of the Web compartment
WebCompartmentDesc	string	A description of the compartment

Examples:

```
ansible-playbook ./001_CreateCompartments_NWMSU.yaml
```

Task Construct

Task Name	Type	Purpose
Get Parent Compartment OCID	command	Attempts to get the OCID of the compartment
Get Parent Compartment	API	Calls API to pull the compartment object from the REST API, ignore on error and fall through to create the parent compartment if not present.
Create the parent compartment	API	Calls the API only if previous API call returns $True for failed
Print Result	debug	Prints the result
Get Parent Compartment	command	Attempts to get the OCID of the compartment, required logic
Get the child compartment OCID if present	command	Attempts to get the OCID of a child compartment
Get child compartment object if present	API	Calls API to pull the compartment object from the REST API, ignore on error and fall through to create the parent compartment if not present.
Create the Compartment	API	Creates the child compartment of the parent, must pass the correct variable inventory objects

Playbook 010_CreateVcns.yaml

Synopsis:

This ANSIBLE playbook creates virtual cloud networks within the tenancy.

Child Playbook: 010_CreateVcn.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name of the VCN
vcn_dnsName	string	The DNS service name that OCI is to assign to the VCN
vcn_cidr	string	The CIDR block for the IP address range to be associated with the VCN
region	string	The datacenter region where the VCN is to be created

Examples:

```
ansible-playbook ./010_CreateVcns.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 010_CreateVcn.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”

Playbook 011_CreateSubnets.yaml

Synopsis:

This ANSIBLE playbook creates virtual cloud networks within the tenancy.

Child Playbook: 011_CreateSubnet.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name of the VCN
sub_name	string	The name to be assigned to the subnet
sub_dns	string	The DNS service name to be assigned by OCI
sub_cidr	string	The CIDR address range to carve out from the VCN. 4 In most cases, this equals the size of the VCN. In cases where multiple subnets are needed, CIDRs are built out along octet boundaries. See RFC 4632
prohibit_public_ip_on_vnic	boolean	In most cases, set to no, which allows public IPs to be bound to VMs. This value should be set to yes when the cloud resources on the subnet should be prevented access to the internet, such as databases and Kubernetes worker nodes.
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./011_CreateSubnets.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 011_CreateSubnet.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”

Playbook 012_CreateLPGs.yaml

Synopsis:

This ANSIBLE playbook creates local peering gateways for each VCN in the tenancy.

Child Playbook: 012_CreateLPG.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name of the VCN
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./012_CreateLPGs.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 012_CreateLPG.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”, along with the correct LPG name you wish to define

Playbook 012_PeerLPGs.yaml

Synopsis:

This ANSIBLE playbook peers gateways for each VCN in the tenancy.

Child Playbook: 012_PeerLPG.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name of the VCN
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./012_PeerLPGs.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 012_CreateLPG.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”, along with the correct LPG names you wish to peer

Playbook 013_CreateNatGateways.yaml

Synopsis:

This ANSIBLE playbook creates NAT gateways for each VCN in the tenancy.

Child Playbook: 013_CreateNatGateway.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name of the VCN
nat_name	string	The name of the NAT to be assigned to the routing appliance
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./013_CreateNatGateways.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 013_CreateNatGateway.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”.

Playbook 015_CreateRouteTables.yaml

Synopsis:

This ANSIBLE playbook creates router tables for each VCN in the tenancy. There are 2 types or router tables created, DRG route tables and standard route tables. DRG route tables are created for routing traffic ingress/egress to the tenancy via the IPSEC tunnel.

Child Playbooks:

015_CreateRouteTable.yaml
015_CreateDRGRouteTable.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
rtb_name	string	The name to be assigned to the route table
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./015_CreateRouteTables.yaml
```

Task Construct

Task Name	Type	Purpose
Get the VCN’s gateway OCID	Command	Gets the OCID of the gateway to default traffic to, must be either a NAT or Internet gateway
include_tasks	Ansible	Calls the child playbook 015_CreateRouteTable.yaml or 015_CreateDRGRouteTable.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”.

Playbook 016_ModifySubnetDefaultRouterTables.yaml

Synopsis:

This ANSIBLE playbook modifies the route table that is assigned to a subnet. This determines how traffic will be routed through the subnet.

Child Playbooks: 016_ModifySubnetDefaultRouterTable.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
sub_name	string	The name of the subnet to apply the route table change to
rtb_name	string	The name of the route table to be assigned to the subnet
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./016_ModifySubnetDefaultRouterTables.yaml
```

Task Construct

Task Name	Type	Purpose
Get the VCN’s gateway OCID	Command	Gets the OCID of the gateway to default traffic to, must be either a NAT or Internet gateway
include_tasks	Ansible	Calls the child playbook 016_ModifySubnetDefaultRouterTable.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”.

Playbook 017_PopulateRouteTables.yaml

Synopsis:

This ANSIBLE playbook purges and re-applies all route table entries in the tenancy.
WARNING! This playbook is disruptive. Run within a VCN Server session when applying.
Do not run during production hours.

Child Playbooks:

017_PopulateRouteTableWithPurge.yaml
017_PopulateRouteTableNoPurge.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
sub_name	string	The name of the subnet to apply the route table change to
rtb_name	string	The name of the route table to be assigned to the subnet
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./017_PopulateRouteTables.yaml
```

Task Construct

Task Name	Type	Purpose
Get NAT or Internet Gateway OCID	command	Gets the OCID for each VCN for the correct default route by subnet
include_tasks	Ansible	Calls the child playbook 017_PopulateRouteTableWithPurge.yaml and applies a purge of all routes with application of the default route to the cloud route table. The route appliance may be either a NAT or internet gateway.
Get OCID for <LPG Name>	command	Each LPG must be obtained for each compartment VCN in order to advertise a route to the LPG. This must be called each time for every route entry to be routed through the connection path between LPGs.
Get the entity object, abort on error	API	Each LPG entity is tested for, abort on error, otherwise proceed to the next task
include_tasks	Ansible	Calls the child playbook 017_PopulateRouteTableNoPurge.yaml. Be certain to pass the correct VAR dictionary object that represents the API required VCN details, along with the correct CIDR to advertise for the connection.
…	…	Repeat for each LPG connection for each VCN

Playbook 018_CreateDynamicRouterGateways.yaml

Synopsis:

This ANSIBLE playbook creates a dynamic routing gateway for the specified VCN within the specified compartment. There may only be one DRG per VCN, but may be multiple DRGs per compartment.

Child Playbooks: 018_CreateDynamicRouterGateway.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
drg_name	string	The name to be assigned to the DRG when created
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./018_CreateDynamicRouterGateways.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 018_CreateDynamicRouterGateway.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”.

Playbook 019_PopulateSecurityLists.yaml

Synopsis:

This ANSIBLE playbook purges and then recreates the access control rules for the VCN's security list. Both ingress and egress rule sets must be purged and reapplied. A purge is only performed with the TCP child playbooks. Always run the purge script and then subsequently call the other scripts without the purge. Please be familiar with security lists and the OCI APIs for managing them prior to modifying this playbook. See https://docs.cloud.oracle.com/en-us/iaas/Content/Network/Concepts/securitylists.htm

WARNING! This script may be disruptive when run. Do not run during business hours.

Child Playbooks:

019_PopulateSecurityOutboundTcpListWithPurge.yaml
019_PopulateSecurityOutboundTcpListNoPurge.yaml
019_PopulateSecurityOutboundIcmpListNoPurge.yaml
019_PopulateSecurityOutboundUdpListNoPurge.yaml
019_PopulateSecurityInboundTcpListWithPurge.yaml
019_PopulateSecurityInboundTcpListNoPurge.yaml
019_PopulateSecurityInboundIcmpListNoPurge.yaml
019_PopulateSecurityInboundUdpListNoPurge.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as "Automation"	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
sec_name	string	The name of the security list to which to apply the changes to
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./019_PopulateSecurityLists.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook for the correct application of a rule. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”. The source, protocol, destination, and port range should be applied as required by applications.

Playbook 020_CreateNetworkSecurityGroups.yaml

Synopsis:

This ANSIBLE playbook creates a network security group for the specified VCN within the specified compartment.

Child Playbooks: 020_CreateNetworkSecurityGroup.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./020_CreateNetworkSecurityGroups.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 018_CreateDynamicRouterGateway.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”. Pass the display name to be assigned to the network security group as a var named “display_name”

Playbook
020_UpdateNetworkSecurityGroupRules.yaml

Synopsis:

This ANSIBLE playbook purges and then recreates the access control rules for the specified security group. Rule purges are not performed, only additions and adjustments. Any NSG rules that are to be removed must be manually deleted using the OCI console. The child playbooks exist only for building ICMP rules at this release. Playbooks are to soon be released for managing UDP and TCP rules.

WARNING! This script may be disruptive when run. Do not run during business hours.

Child Playbooks:

020_UpdateNsgEgressIcmpRule.yaml
020_UpdateNsgIngressIcmpRule.yaml
020_UpdateNetworkSecurityGroup.py

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
region	string	The region where the VCN was created in.

Examples:

```
ansible-playbook ./020_UpdateNetworkSecurityGroupRules.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook for the correct application of a rule. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”. The source, protocol, destination, and port range should be applied as required by applications.

Playbook 100_AssignVMsToNSGs.yaml

Synopsis:

This ANSIBLE playbook assigns a VM instance's primary network interface to the specified network security group.

Child Playbooks: 100_AssignVmToNsg.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as "Automation"	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
subnet_name	string	The name of the subnet the VM instance is associated with
region	string	The region where the VCN was created in.
VM Dictionary object name, such as DKCUDCP01	dictionary	
instance_hostname	string	The name of the VM instance to apply the NSG to
nsg_name	string	The name of the NSG to apply to the VM instance network interface

Examples:

```
ansible-playbook ./020_UpdateNetworkSecurityGroupRules.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child 100_AssignVmToNsg.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as "Automation".

Playbook 100_CreateVMs.yaml

Synopsis:

This ANSIBLE playbook creates a VM instance if not already present.

Child Playbooks:

100_CreateLinuxVm.yaml
100_CreateWindowsVm.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
subnet_name	string	The name of the subnet the VM instance is associated with
region	string	The region where the VCN was created in.
VM Dictionary object name, such as DKCUDCP01	dictionary	
instance_hostname	string	The name of the VM instance to apply the NSG to
assign_public_ip	boolean	Set to true if VM instance is send or receive traffic outside of tenancy to the internet, set to false if VM instance is to not send or receive traffic from outside of OCI to the internet.
availability_domain	string	The availability domain to create the VM in. Get availability domain strings with ‘oci iam availability-domain list’
private_ip	string	The private IP address to adminster to the VM instance
shape	string	The shape to apply to the VM
ssh_authorized_keys	string or list of strings	The SSH keys or key that will allow ssh login access to the VM instance. Recommend using the Ansible user public key. The golden image file should already have keys & identities embedded in it. This is a field for Linux VM instances only. Omit this value for Windows VM instances.

Parameter	Value	Comments
images ole_golden_image_id windows_golden_image Other Var Names	string	The OCID of an image from which to clone the VM instance

Examples:

```
ansible-playbook ./100_CreateVMs.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child 100_CreateLinuxVm.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”. Pass a var called boot_volume_size_in_gbs with an integer value of 50 or greater for the boot vol initial size, 100 recommended, pass a var image_id that passes one of the previously defined image OCID vars.

Playbook 101_CreateAndAttachBlockVolumes.yaml

Synopsis:

This ANSIBLE playbook creates and attaches a block volume to the specified VM instance.

Child Playbooks: 101_CreateAndAttachBlockVolume.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name to be assigned to the VCN
rsubnet_name	string	The name of the subnet the VM is assigned to
region	string	The region where the VCN was created in.
VM instance name VAR, such as DKCURMANP01	dictionary	
instance_hostname	string	The name of the VM instance
availability_domain	string	The availability domain where the VM was created in. Use GetVm.ps1 to get details regarding the VM instance’s availability domain.
DataVolSize	integer	The size of the data volume, in GB, minimum value 50

Examples:

```
ansible-playbook ./101_CreateAndAttachBlockVolumes.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 101_CreateAndAttachBlockVolume.yaml. Be certain to pass to the playbook the correct dictionary object defined in the YAML, such as “Automation”.Pass as VM the dictionary object for the VM instance

Playbook 110_CreateVolBackupPolicies.yaml

Synopsis:

This ANSIBLE playbook creates a volume backup policy.

Child Playbooks:

110_CreateVolBackupPolicy.yaml
110_CreateVolBackupPolicy.py

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
VMs	Array	A list of objects with the values required to create the backup policies
VM	dictionary	A list of one or more objects
compartment	string	The compartment to create the policy in
cross_region_backup	*string*	*[depreciated] Limitations in the number of cross tenancy parallel copies exist, we do not recommend setting this within a policy*
region	string	The datacenter region where the policy will be stored
instance_hostname	string	The name of the VM instance for which this policy exists
instance_compartment	string	The compartment where the instance resides in
policy_name	string	The name of the policy to create
backup_start_time	integer	The start time in hour for the backup to begin, valid entries between 0 and 23
backup_retention	integer	The number of days to retain the backup

Examples:

```
ansible-playbook ./101_CreateAndAttachBlockVolumes.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 110_CreateVolBackupPolicy.yaml recursively using the array VMs

Playbook 120_AssignVolBackupPolicies.yaml

Synopsis:

This ANSIBLE playbook assigns a backup policy to both the boot and data volume of a VM instance.

Child Playbooks:

120_AssignBootVolBackupPolicy.yaml
120_AssignBlockVolBackupPolicy.yaml
110_UpdateVolBackupPolicy.py

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
VMs	Array	A list of objects with the values required to create the backup policies
VM	dictionary	A list of one or more objects
compartment	string	The compartment to create the policy in
cross_region_backup	string	[depreciated] Limitations in the number of cross tenancy parallel copies exist, we do not recommend setting this within a policy
region	string	The datacenter region where the policy will be stored
instance_hostname	string	The name of the VM instance for which this policy exists
instance_compartment	string	The compartment where the instance resides in
policy_name	string	The name of the policy to create
backup_start_time	integer	The start time in hour for the backup to begin, valid entries between 0 and 23
backup_retention	integer	The number of days to retain the backup

Examples:

```
ansible-playbook 120_AssignVolBackupPolicies.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 120_AssignBootVolBackupPolicy.yaml and 120_AssignBlockVolBackupPolicy.yaml, recursively using the array VMs

Playbook 200_CreateDBaaSs.yaml

Synopsis:

This ANSIBLE playbook assigns a backup policy to both the boot and data volume of a VM instance.

Child Playbooks:

200_CreateDBaaS.yaml
200_CreateDBaaS.py

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
Var name for VCN properties, such as “Automation”	dictionary	Object for Automation VCN
compartment	string	Name of the compartment to hold the VCN
vcn_name	string	The name of the VCN
subnet_name	string	The name of the subnet the DB System will be assigned to
region	string	The region where the VCN was created in.
DB System Name, such as DKCUDBP01	dictionary	
availability_domain	string	The availability domain where the DB System is to be created. Get a list with ‘oci iam availability-domain list’
admin_password	string	The initial password to assign to the compartment and the database, see https://docs.oracle.com/en/database/oracle/oracle-database/18/ntdbi/requirements-for-database-password.html#GUID-FCB0D571-E16D-4CC0-8B39-AD06BD94739D
auto_backup_enabled	boolean	Set to False for not enabling the OCI database backup service, or True to enable. We recommend that auto backup not be enabled when initially standing up a new DB System
database_edition	string	The edition of the Oracle license to deploy. See https://docs.cloud.oracle.com/en-us/iaas/Content/Database/Concepts/overview.htm

Parameter	Value	Comments
data_storage_percentage	integer	The amount of storage in the aggregate that will be usable to Oracle Database. We recommend using the default value of 80 for 80%.
db_display_name	string	The display name for the database. We recommend this be the same as the container database name.
db_name	string	The name for the database. We recommend this be the same as the container database name.
db_system_cpu_core_count	integer	Not used for virtual machine databases, leave at a value of 1 in all cases
db_storage_mgnt	string	Use ASM for grid control managed storage, or LVM for logical file system managed storage. We recommend ASM in all cases.
db_version	string	Enter the correct version for the initial Oracle Home.
db_workload	string	May be either “OLTP” for transactional or “DSS” for warehouse.
disk_redundancy	string	May be “NORMAL” for standard disk mirroring, or “HIGH” for triple mirror redundancy
display_name	string	The name for the DB System. We recommend this be the same as the container database name.
hostname	string	The host name to be assigned to the service node
initial_data_storage_size_in_gb	integer	The initial size of usable storage, starts at 256
license_model	string	May be “LICENSE_INCLUDED” for including Oracle Database licenses in the cloud, or “BRING_YOUR_OWN_LICENSE” for eligible processor licenses that may be moved to the cloud
ncharacter_set	string	The character set to be applied to the Oracle Home. We recommend using the default of “AL16UTF16”
node_count	integer	May be 1 or 2. Determines the number of service nodes that will be deployed with the DB System. Use 1 in all cases where Oracle RAC is not licensed. Use 2 in cases where Oracle RAC is licensed.
pdb_name	string	The private database, or instance name
shape	string	The VM instance shape of the service node(s) that will be bound to the DB System
ssh_public_keys	string	The path to the initial SSH public key that will be applied to the DB System
time_zone	string	The UTC defined timezone that will be applied to the DB System

Examples:

```
ansible-playbook 200_CreateDBaaSs.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 200_CreateDBaaS.yaml. Be certain to provide the VAR dictionary objects for the correct DB System and VCN

Playbook 300_CreateFileSystems.yaml

Synopsis:

This ANSIBLE playbook creates a file system appliance within the tenancy for the specified VCN and subnet.

Child Playbooks: 300_CreateFileSystem.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
File system name, such as “DKCUFST01”	dictionary	
availability_domain	string	The availability domain where the file system will be created
display_name	string	The name that will be assigned to the file system
mt_display_name	string	The name that will be assigned to the filesystem mount point
hostname_label	string	The name that will be assigned to the underlying host that runs the appliance, may be the same as the display name
ip_address	string	The IP address to be assigned to the file system. Must be within range of the subnet CIDR that the file system will be assigned to
VCN Name	dictionary	
compartment	string	The name of the compartment where the VCN is
vcn_name	string	The name of the VCN
subnet_name	string	The name of the subnet
region	string	The datacenter region where the VCN was created in

Examples:

```
300_CreateFileSystems.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 300_CreateFileSystem.yaml. Be certain to provide the VAR dictionary objects for the correct File System and VCN

Playbook 310_ManageExports.yaml

Synopsis:

This ANSIBLE playbook creates a file system appliance export within the tenancy for the specified VCN and subnet. It also manages the export permissions list post creation.

Child Playbooks:

310_CreateExport.yaml
310_ManageExportWithPurge.yaml
310_ManageExportNoPurge.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
File system name, such as “DKCUFST01”	dictionary	
availability_domain	string	The availability domain where the file system will be created
display_name	string	The name that will be assigned to the file system
mt_display_name	string	The name that will be assigned to the filesystem mount point
hostname_label	string	The name that will be assigned to the underlying host that runs the appliance, may be the same as the display name
ip_address	string	The IP address to be assigned to the file system. Must be within range of the subnet CIDR that the file system will be assigned to
VCN Name	dictionary	
compartment	string	The name of the compartment where the VCN is
vcn_name	string	The name of the VCN
subnet_name	string	The name of the subnet
region	string	The datacenter region where the VCN was created in

Examples:

```
ansible-playbook 310_ManageExports.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 310_CreateExport.yaml. Be certain to provide the VAR dictionary objects for the correct File System and VCN, provide the value for the path, which will be the remote mountable path for NFS clients., the access node, such as “READ_WRITE”, and identity_squash value, use “NONE” if non-applicable.
include_tasks	Ansible	Calls the child playbook 310_ManageExportWithPurge.yaml. Be certain to provide the VAR dictionary objects for the correct File System and VCN, provide the value for the path, which will be the remote mountable path for NFS clients., the access node, such as “READ_WRITE”, and identity_squash value, use “NONE” if non-applicable. Provide the NFS client IP address that will require access permissions.
include_tasks	Ansible	Calls the child playbook 310_ManageExportNoPurge.yaml. Be certain to provide the VAR dictionary objects for the correct File System and VCN, provide the value for the path, which will be the remote mountable path for NFS clients., the access node, such as “READ_WRITE”, and identity_squash value, use “NONE” if non-applicable. Provide the NFS client IP address that will require access permissions.

Playbook 400_CreateKbClusters.yaml

Synopsis:

This ANSIBLE playbook creates a Kubernetes cluster service within the tenancy for the specified VCN and subnet.

Child Playbooks: 400_CreateKbCluster.yaml

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
python_path	string	The fully qualified path and file name to the python interpreter. Must be version 3.6 or later of python
Cluster service name, such as “DKCUKBC01”	dictionary	
kubernetes_version	string	The version of the Kubernetes service to deploy
name	string	The name of the Kubernetes service
options	dictionary	
add_ons	dictionary	
is_kubernetes_dashboard_enabled	boolean	Set to yes, always enable the dashboard service
is_tiller_enabled	boolean	Set to yes to enable this feature, no to disable it. The default is no
kubernetes_network_config	dictionary	
pods_cidr	string	The private CIDR for the worker nodes, default is 10.96.0.0/16
services_cidr	string	The private CIDR for the service. The default is 20.244.0.0/16
wait	boolean	Set to yes to wait for the cluster service to become available prior to exiting, or no to not wait
VCN Name	dictionary	
compartment	string	The compartment where the VCN was created in
vcn_name	string	The VCN name

Parameter	Value	Comments
subnet_name	string	The name of the subnet the Kubernetes service will be assigned to
region	string	The datacenter region where the VCN was created in

Examples:

```
ansible-playbook 400_CreateKbClusters.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 400_CreateKbClusters.yaml. Be certain to provide the VAR dictionary objects for the correct Kubernetes cluster and VCN.

Playbook 410_CreateKbNodePools.yaml

Synopsis:

This ANSIBLE playbook creates a Kubernetes node pool within the tenancy for the specified Kubernetes cluster service.

Child Playbooks:

410_CreateKbNodePool.yaml
410_CreateNodePool.py

Parameters:

Parameter	Value	Comments
gather_facts	boolean	Always set to no
hosts	string	Always set to localhost
python_path	string	The fully qualified path and file name to the python interpreter. Must be version 3.6 or later of python
Nodepool name	dictionary	
kubernetes_version	string	The version of Kubernetes to deploy to the nodepool. Should match the version deployed in the cluster service
name	string	The name of the Kubernetes cluster service to bind this node pool to
Nodepool	dictionary	
name		The name of the nodepool cluster
node_image_name	string	The name of the Oracle Linux image to deploy to the cluster worker nodes
node_shape	string	The VM instance shape to apply to the cluster worker nodes
quantity_per_subnet	integer	The number of nodes to create per subnet. Must be at least 3, and may be increased in increments of 3
AD1	string	The name of the first availability domain
AD2	string	The name of the second availability domain
AD3	string	The name of the third availability domain
ssh_key	string	The public SSH key that will be applied to the cluster worker nodes
VCN Name	dictionary	

Parameter	Value	Comments
compartment	string	The compartment where the VCN was created in
vcn_name	string	The VCN name
subnet_name	string	The name of the subnet the Kubernetes service will be assigned to
pool_subnet_name	string	The subnet that the Kubernetes worker nodes will be assigned to
region	string	The datacenter region where the VCN was created in

Examples:

```
ansible-playbook 410_CreateKbNodePools.yaml
```

Task Construct

Task Name	Type	Purpose
include_tasks	Ansible	Calls the child playbook 410_CreateKbNodePool.yaml. Be certain to provide the VAR dictionary objects for the correct Kubernetes nodepool and VCN.