

Oracle® Enterprise Manager Ops Center
Cloud Infrastructure API and CLI Reference Guide
12c Release 1 (12.1.3.0.0)
E25150-05

March 2013

Oracle Enterprise Manager Ops Center Cloud Infrastructure API and CLI Reference Guide, 12c Release 1 (12.1.3.0.0)

E25150-05

Copyright © 2007, 2013, Oracle and/or its affiliates. All rights reserved.

Primary Author: Karen Orozco

Contributing Author: Luis Ramos

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xv
Audience	xv
Documentation Accessibility	xv
Related Documents	xv
Conventions	xv
1 Introduction	
Overview of vDC Management	1-1
Important Concepts	1-2
How APIs and CLI Work	1-2
Synchronous Semantic	1-3
Overview of the Functionality	1-3
2 Getting Started	
Prerequisites	2-1
Installation	2-1
Cloud Infrastructure API	2-1
Cloud Infrastructure Java Client API	2-1
Cloud Infrastructure CLI	2-2
Next Steps	2-3
3 Managing Resources	
Overview of Cloud User Resources	3-1
Status of Account Resources	3-2
Managing Account Access	3-3
Creating an Access Key	3-4
Viewing Access Key Information	3-4
Deleting an Access Key	3-5
Viewing vDC Capabilities	3-5
Managing Virtual Networks	3-6
Creating a vNet	3-6
Deleting a vNet	3-7
Viewing vNet Information	3-7
Managing vIP Addresses	3-8
Allocating a vIP Address	3-8

Deallocating a vIP Address	3-8
Viewing vIP Addresses	3-9
Managing Server Templates	3-9
Uploading a Server Template from a vServer	3-10
Uploading a Server Template from a URL	3-10
Uploading a Server Template from an Assembly	3-11
Deleting a Server Template	3-12
Viewing Server Template Information	3-12
Managing Storage	3-13
Creating a Volume	3-13
Deleting a Volume	3-14
Viewing Volume Information	3-14
Attaching a Volume to a vServer	3-15
Detaching a Volume from a vServer	3-15
Importing a Volume	3-16
Creating a Snapshot	3-16
Deleting a Snapshot	3-17
Viewing Snapshot Information	3-17
Managing Distribution Groups	3-18
Creating a Distribution Group	3-18
Deleting a Distribution Group	3-19
Viewing Distribution Group Information	3-19
Managing vServers	3-20
Viewing vServer Types	3-20
Creating a vServer	3-21
Deleting a vServer	3-22
Viewing vServer Information	3-22
Viewing vServer Metrics	3-23
Stopping a vServer	3-24
Starting a vServer	3-24
Rebooting a vServer	3-25
Sending a Message to a vServer	3-25
Receiving a Message from a vServer	3-25
Managing Key Pairs	3-26
Creating a Key Pair	3-26
Importing a Key Pair	3-27
Deleting a Key Pair	3-27
Viewing Key Pair Information	3-28
Managing Attributes of an Account Resource	3-28
Viewing Attributes of an Account Resource	3-28
Modifying Attributes of an Account Resource	3-29
Managing Tags	3-29
Creating a Tag	3-30
Removing a Tag	3-30
Viewing Tag Information	3-30

4 Cloud Infrastructure API Reference

Overview of the Cloud Infrastructure API	4-1
Cloud Infrastructure API Requests	4-1
Required Web Service HTTP Request Parameters	4-2
Signing a Web Service Request	4-2
List of Actions	4-4
AllocateIpAddressesRequest.....	4-5
AttachVolumesToVserverRequest	4-7
CreateAccessKeyAsObjectRequest.....	4-9
CreateAccessKeyToFileRequest.....	4-10
CreateDistributionGroupRequest.....	4-12
CreateKeyPairAsObjectRequest.....	4-14
CreateKeyPairToFileRequest.....	4-15
CreateSnapshotRequest.....	4-16
CreateTagsRequest.....	4-18
CreateVnetRequest.....	4-20
CreateVolumeRequest.....	4-22
DeleteAccessKeyRequest	4-24
DeleteDistributionGroupRequest	4-25
DeleteKeyPairRequest	4-26
DeleteSnapshotRequest.....	4-27
DeleteTagsRequest.....	4-28
DeleteVnetRequest.....	4-30
DeleteVolumeRequest	4-31
DeregisterServerTemplateRequest	4-32
DescribeAccessKeysRequest	4-33
DescribeAccountsRequest.....	4-35
DescribeAttributesRequest	4-37
DescribeDistributionGroupsRequest	4-39
DescribeIpAddressesRequest.....	4-41
DescribeKeyPairsRequest	4-43
DescribeServerTemplatesRequest	4-45
DescribeSnapshotsRequest	4-47
DescribeTagsRequest.....	4-49
DescribeVdcCapabilitiesRequest	4-51
DescribeVnetsRequest	4-53
DescribeVolumesRequest	4-55
DescribeVserverMetricsRequest	4-57
DescribeVserversRequest.....	4-59
DescribeVserverTypesRequest.....	4-62
DetachVolumesFromVserverRequest	4-63
ImportKeyPairRequest	4-65
ImportVolumeRequest	4-67
ModifyAttributesRequest	4-69
RebootVserversRequest.....	4-71
ReceiveMessageFromVserverRequest	4-72
RegisterAccessKeyRequest.....	4-74

RegisterKeyPairRequest.....	4-76
RegisterServerTemplateFromUrlRequest.....	4-78
RegisterServerTemplateFromVserverRequest.....	4-80
RegisterServerTemplatesFromAssemblyRequest.....	4-82
ReleaseIpAddressesRequest.....	4-84
RunVserverRequest.....	4-85
RunVserversRequest.....	4-88
SendMessageToVserverRequest.....	4-91
StartVserversRequest.....	4-93
StopVserversRequest.....	4-94
TerminateVserversRequest.....	4-96
Data Types	4-98
AccountIdType.....	4-99
DescribeAccessKeysResultItem.....	4-100
DescribeAccountsResultItem.....	4-101
DescribeDistributionGroupsResultItem.....	4-102
DescribeIpAddressesResultItem.....	4-103
DescribeKeyPairsResultItem.....	4-104
DescribeServerTemplatesResultItem.....	4-105
DescribeSnapshotsResultItem.....	4-106
DescribeTagsResultItem.....	4-107
DescribeVdcCapabilitiesResultItem.....	4-108
DescribeVnetsResultItem.....	4-109
DescribeVolumesResultItem.....	4-110
DescribeVserverMetricsResultItem.....	4-111
DescribeVserversResultItem.....	4-112
DescribeVserverTypesResultItem.....	4-114
DistributionGroupIdType.....	4-115
FilterItem.....	4-116
GenericString.....	4-117
HostnameType.....	4-118
Message.....	4-119
PositiveInteger.....	4-120
ResourceAttribute.....	4-121
ResourceIdType.....	4-122
SnapshotIdType.....	4-123
ServerTemplateIdType.....	4-124
TagType.....	4-125
URLType.....	4-126
VnetIdType.....	4-127
VolumeIdType.....	4-128
VserverIdType.....	4-129

5 Cloud Infrastructure CLI Reference

Common Command Options	5-1
List of Commands	5-3
akm-create-access-key.....	5-4

akm-delete-access-key	5-7
akm-describe-access-keys	5-10
akm-describe-accounts	5-13
iaas-allocate-ip-addresses	5-16
iaas-attach-volumes-to-vserver	5-19
iaas-create-distribution-group.....	5-21
iaas-create-key-pair.....	5-24
iaas-create-server-template-from-assembly	5-26
iaas-create-server-template-from-url	5-28
iaas-create-server-template-from-vserver	5-30
iaas-create-snapshot.....	5-32
iaas-create-tags	5-35
iaas-create-vnet.....	5-37
iaas-create-volume	5-40
iaas-delete-distribution-group	5-43
iaas-delete-key-pair.....	5-45
iaas-delete-server-template.....	5-47
iaas-delete-snapshot.....	5-49
iaas-delete-tags	5-51
iaas-delete-vnet.....	5-53
iaas-delete-volume	5-55
iaas-describe-attributes.....	5-57
iaas-describe-distribution-groups.....	5-60
iaas-describe-ip-addresses	5-62
iaas-describe-key-pairs	5-64
iaas-describe-server-templates.....	5-67
iaas-describe-snapshots.....	5-70
iaas-describe-tags	5-72
iaas-describe-vdc-capabilities.....	5-75
iaas-describe-vnets	5-77
iaas-describe-volumes	5-80
iaas-describe-vserver-metrics.....	5-83
iaas-describe-vservers.....	5-86
iaas-describe-vserver-types	5-89
iaas-detach-volumes-from-vserver.....	5-91
iaas-import-key-pair	5-93
iaas-import-volume.....	5-95
iaas-modify-attributes	5-97
iaas-reboot-vservers.....	5-100
iaas-receive-message-from-vserver	5-102
iaas-release-ip-addresses.....	5-104
iaas-run-vserver.....	5-107
iaas-run-vservers	5-110
iaas-send-messages-to-vserver.....	5-113
iaas-start-vservers	5-115
iaas-stop-vservers.....	5-117
iaas-terminate-vservers	5-119

A Web Service Client Example

Description of Web Service API Requests	A-1
AKM Request.....	A-1
REQUEST_DATA Specification.....	A-2
IAAS Request.....	A-2
REQUEST_IAAS_DATA Specification	A-2
SIGNATURE_BLOCK Specification	A-3
IAAS Data Used for Signature	A-3
Sending a Web Service Request	A-4
Using the Web Browser.....	A-4
Using WGET Utility.....	A-4
Creating Public and Private Keys	A-5
Private Key	A-5
Public Key	A-5
Overview of the WebUtil Application	A-5
How the WebUtil Application Works.....	A-8
Web Service Request Examples	A-9
Creating an Access Key	A-9
Executing actions for an account	A-10
Viewing All Available Virtual Networks.....	A-11
Viewing Specific Virtual Networks by Using IDs	A-12
Viewing Specific Virtual Networks by Using Filters.....	A-13
Creating a Private Virtual Network	A-14

Index

List of Examples

3-1	Creating an access key using the Cloud Infrastructure API.....	3-4
3-2	Creating an access key using the Cloud Infrastructure CLI.....	3-4
3-3	Viewing access key information using the Cloud Infrastructure API.....	3-5
3-4	Viewing access key information using the Cloud Infrastructure CLI.....	3-5
3-5	Deleting an access key information using the Cloud Infrastructure API.....	3-5
3-6	Deleting an access key information using the Cloud Infrastructure CLI.....	3-5
3-7	Viewing vDC capabilities using the Cloud Infrastructure API.....	3-5
3-8	Viewing vDC capabilities using the Cloud Infrastructure CLI.....	3-6
3-9	Creating a vNet using the Cloud Infrastructure API.....	3-6
3-10	Creating a vNet using the Cloud Infrastructure CLI.....	3-6
3-11	Deleting a vNet using the Cloud Infrastructure API.....	3-7
3-12	Deleting a vNet using the Cloud Infrastructure CLI.....	3-7
3-13	Viewing vNet information using the Cloud Infrastructure API.....	3-7
3-14	Viewing vNet information using the Cloud Infrastructure CLI.....	3-7
3-15	Allocating a vIP address using the Cloud Infrastructure API.....	3-8
3-16	Allocating a vIP address using the Cloud Infrastructure CLI.....	3-8
3-17	Deallocating a vIP address using the Cloud Infrastructure API.....	3-8
3-18	Deallocating a vIP address using the Cloud Infrastructure CLI.....	3-9
3-19	Viewing vIP addresses using the Cloud Infrastructure API.....	3-9
3-20	Viewing vIP addresses using the Cloud Infrastructure CLI.....	3-9
3-21	Uploading a server template from a vServer using the Cloud Infrastructure API.....	3-10
3-22	Uploading a server template from a vServer using the Cloud Infrastructure CLI.....	3-10
3-23	Uploading a server template from a URL using the Cloud Infrastructure API.....	3-11
3-24	Uploading a server template from a URL using the Cloud Infrastructure CLI.....	3-11
3-25	Uploading a server template from an assembly using the Cloud Infrastructure API...	3-11
3-26	Uploading a server template from an assembly using the Cloud Infrastructure CLI...	3-11
3-27	Deleting a server template using the Cloud Infrastructure API.....	3-12
3-28	Deleting a server template using the Cloud Infrastructure CLI.....	3-12
3-29	Viewing server template information using the Cloud Infrastructure API.....	3-13
3-30	Viewing server template information using the Cloud Infrastructure CLI.....	3-13
3-31	Creating a volume using the Cloud Infrastructure API.....	3-13
3-32	Creating a volume using the Cloud Infrastructure CLI.....	3-14
3-33	Deleting a volume using the Cloud Infrastructure API.....	3-14
3-34	Deleting a volume using the Cloud Infrastructure CLI.....	3-14
3-35	Viewing volume information using the Cloud Infrastructure API.....	3-14
3-36	Viewing volume information using the Cloud Infrastructure CLI.....	3-15
3-37	Attaching a volume to a vServer using the Cloud Infrastructure API.....	3-15
3-38	Attaching a volume to a vServer using the Cloud Infrastructure CLI.....	3-15
3-39	Detaching a volume to a vServer using the Cloud Infrastructure API.....	3-15
3-40	Detaching a volume to a vServer using the Cloud Infrastructure CLI.....	3-16
3-41	Importing a volume using the Cloud Infrastructure API.....	3-16
3-42	Importing a volume using the Cloud Infrastructure CLI.....	3-16
3-43	Creating a snapshot using the Cloud Infrastructure API.....	3-17
3-44	Creating a snapshot using the Cloud Infrastructure CLI.....	3-17
3-45	Deleting a snapshot using the Cloud Infrastructure API.....	3-17
3-46	Deleting a snapshot using the Cloud Infrastructure CLI.....	3-17
3-47	Viewing snapshot information using the Cloud Infrastructure API.....	3-18
3-48	Viewing snapshot information using the Cloud Infrastructure CLI.....	3-18
3-49	Creating a distribution group using the Cloud Infrastructure API.....	3-18
3-50	Creating a distribution group using the Cloud Infrastructure CLI.....	3-18
3-51	Deleting a distribution group using the Cloud Infrastructure API.....	3-19
3-52	Deleting a distribution group using the Cloud Infrastructure CLI.....	3-19
3-53	Viewing distribution groups information using the Cloud Infrastructure API.....	3-19
3-54	Viewing distribution groups information using the Cloud Infrastructure CLI.....	3-20

3-55	Listing vServer types using the Cloud Infrastructure API	3-21
3-56	Listing vServer types using the Cloud Infrastructure CLI	3-21
3-57	Creating a vServer using the Cloud Infrastructure API.....	3-22
3-58	Creating a vServer using the Cloud Infrastructure CLI.....	3-22
3-59	Deleting a vServer using the Cloud Infrastructure API.....	3-22
3-60	Deleting a vServer using the Cloud Infrastructure CLI	3-22
3-61	Viewing vServer information using the Cloud Infrastructure API.....	3-23
3-62	Viewing vServer information using the Cloud Infrastructure CLI	3-23
3-63	Viewing vServer metrics using the Cloud Infrastructure API.....	3-24
3-64	Viewing vServer metrics using the Cloud Infrastructure CLI	3-24
3-65	Stopping a vServer using the Cloud Infrastructure API.....	3-24
3-66	Stopping a vServer using the Cloud Infrastructure CLI	3-24
3-67	Starting a vServer using the Cloud Infrastructure API.....	3-24
3-68	Starting a vServer using the Cloud Infrastructure CLI	3-25
3-69	Rebooting a vServer using the Cloud Infrastructure API.....	3-25
3-70	Rebooting a vServer using the Cloud Infrastructure CLI	3-25
3-71	Sending messages to a vServer using the Cloud Infrastructure API	3-25
3-72	Sending messages to a vServer using the Cloud Infrastructure CLI.....	3-25
3-73	Receiving message from a vServer using the Cloud Infrastructure API.....	3-26
3-74	Receiving message from a vServer using the Cloud Infrastructure CLI	3-26
3-75	Creating a key pair using the Cloud Infrastructure API.....	3-26
3-76	Creating a key pair using the Cloud Infrastructure CLI	3-27
3-77	Importing a key pair using the Cloud Infrastructure API.....	3-27
3-78	Importing a key pair using the Cloud Infrastructure CLI	3-27
3-79	Deleting a key pair using the Cloud Infrastructure API.....	3-27
3-80	Deleting a key pair using the Cloud Infrastructure CLI	3-28
3-81	Viewing key pair information using the Cloud Infrastructure API.....	3-28
3-82	Viewing key pair information using the Cloud Infrastructure CLI	3-28
3-83	Viewing attributes of an account resource using the Cloud Infrastructure API.....	3-29
3-84	Viewing attributes of an account resource using the Cloud Infrastructure CLI	3-29
3-85	Modifying vNet attributes using the Cloud Infrastructure API	3-29
3-86	Modifying vNet attributes using the Cloud Infrastructure CLI	3-29
3-87	Creating a tag using the Cloud Infrastructure API.....	3-30
3-88	Creating a tag using the Cloud Infrastructure CLI.....	3-30
3-89	Deleting a tag using the Cloud Infrastructure API.....	3-30
3-90	Deleting a tag using the Cloud Infrastructure CLI	3-30
3-91	Viewing tag information using the Cloud Infrastructure API.....	3-31
3-92	Viewing tag information using the Cloud Infrastructure CLI	3-31
4-1	URL for an HTTP GET request to the IAAS servlet:.....	4-3
4-2	URL for an HTTP GET request to the AKM servlet:	4-3
A-1	A Complete AKM Request	A-2
A-2	Sample REQUEST_IAAS_DATA Specification	A-3
A-3	Sample SIGNATURE_BLOCK Specification	A-3
A-4	Example of the IAAS Data to Sign.....	A-4
A-5	Java Representation of the Data to Sign	A-4

List of Figures

1-1	Cloud Infrastructure APIs and CLI Execution Flow	1-2
-----	--	-----

List of Tables

1-1	vDC Management	1-3
1-2	Account Management	1-3
1-3	Server Template Management	1-3
1-4	vServer Management.....	1-4
1-5	Volume Management	1-4
1-6	Snapshot Management.....	1-4
1-7	Virtual IP Management.....	1-4
1-8	Virtual Network Management.....	1-4
1-9	Account and vServer Access Management.....	1-4
1-10	Distribution Groups Management	1-5
1-11	Tag Management	1-5
4-1	Required Parameters for an HTTP Request.....	4-2
5-1	Common Command Options.....	5-1

Preface

This guide covers installation procedures and operation reference material for the cloud infrastructure API and cloud infrastructure CLI of Oracle Enterprise Manager Ops Center.

Audience

This document is intended for advanced users or developers who require access to manage the virtual servers, storage, and network infrastructure as a service from a shell or scripts.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the Oracle Enterprise Manager Ops Center documentation library.

Oracle Enterprise Manager Ops Center provides online help. Click Help at the top-right corner of any page in the user interface to display the online help window.

For the latest releases of this and other Oracle documentation, check the Oracle Technology Network at

<http://www.oracle.com/technetwork/documentation/index.html#em>

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

Oracle Enterprise Manager Ops Center provides APIs and a command-line interface (CLI) to enable access to a subset of the virtual datacenter (vDC) functionality provided to cloud users.

These APIs and CLI offer a way for cloud users to programmatically manage allocated virtual resources in a vDC account, providing the ability to create and manage virtual servers (vServers) to deploy applications in an infrastructure as a service (IaaS) mode.

This guide provides information about:

- Cloud infrastructure API – A programmatic Web service interface for managing virtual datacenter (vDC) resources allocated in vDC accounts. This API is also referred to simply as Web service in this guide.
- Cloud infrastructure Java client API – A Java client API for interacting with the Web service interface. This API is also referred to simply as Java client API in this guide.
- Cloud infrastructure CLI – A Java-based command-line interface that wraps the cloud infrastructure Java client API with simple scripts. It is also referred to simply as CLI in this guide.

The following topics are covered in this chapter:

- [Overview of vDC Management](#)
- [How APIs and CLI Work](#)
- [Overview of the Functionality](#)

Overview of vDC Management

Oracle Enterprise Manager Ops Center supports cloud management functions through its vDC management system in three main areas:

- vDC infrastructure administration – Area related to the management and allocation of physical resources for the vDC by a cloud administrator. The managed resources are the server pools, storage, and networks.
- Cloud users management – Area related to the management of the cloud users, their linkage with the authentication systems, and their association with the vDC accounts that they are authorized to access by a cloud administrator.
- vDC resource utilization – Area related to cloud user tasks to perform lifecycle management of vServers and resources in a vDC account. There might be more than one cloud user associated with an account with the same level of privileges and access control.

See *Oracle Enterprise Manager Ops Center Feature Reference Guide* for more information about vDC management.

Important Concepts

The following list describes vDC management concepts used in this guide:

- Virtual datacenter – Consolidation of virtualization servers, storage, and network resources to be used optimally and securely for mixed and dynamic workloads.
- Account – An account entitles designated cloud users the right to use computing, network, and storage resources of vDC. The account provides the required capabilities to manage these resources. The amount of virtual CPU (vCPU), memory, and storage resources that can be used from the available vDC resources.
- Cloud administrator – An Oracle Enterprise Manager Ops Center user with the cloud administrator role. A cloud administrator can create and manage the vDCs, accounts, and cloud users.
- Cloud user – An Oracle Enterprise Manager Ops Center user with the cloud user role. Cloud users can have access to different accounts to manage the allocated resources. Cloud users can create virtual servers to host or access applications.

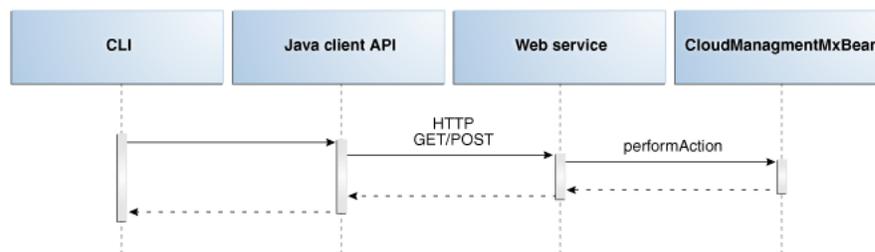
How APIs and CLI Work

For this release, cloud infrastructure APIs and CLI support the management of the resources in vDC accounts by using the following virtualization technologies:

- Oracle Solaris Zones
- Oracle VM Server for x86

Figure 1–1 illustrates the execution flow and components that support the cloud infrastructure APIs and CLI.

Figure 1–1 Cloud Infrastructure APIs and CLI Execution Flow



- The cloud infrastructure CLI is delivered as a standalone package and can be used without an Oracle Enterprise Manager Ops Center installation in the computer where CLI is deployed. This element offers the same functionality as the cloud infrastructure API and Java client API.
- The cloud infrastructure Java client API provides for the cloud infrastructure CLI a common set of functionality for executing actions through the cloud infrastructure API. The Java client API is delivered as a standalone package.
- The cloud infrastructure API is a Web service that exposes a subset of the functionality of the vDC management system. A cloud infrastructure API request triggers an action by calling the corresponding action of the vDC management system.

The functionality exposed by the Web service can then be accessed programmatically using the Java Client API, the CLI, or by making calls directly to the Web service.

Synchronous Semantic

The cloud infrastructure API provides a synchronous call semantic for all modify or create actions. The call returns at least the identifiers of the changed or created resources.

The creation of the resource IDs and the mapping to jobs is handled by the vDC management system. The cloud infrastructure API must wait actively for notifications from the Job Manager to get the IDs of the changed or created resources. Jobs are specially tailored to deliver fast results; the cloud infrastructure API waits only until the necessary results are available.

The cloud infrastructure API returns a valid identifier even when resource creation fails. A cloud user must be able to view resource information, such as name, description, and status using the identifier returned. For more information, see "[Status of Account Resources](#)".

Overview of the Functionality

The vDC management system in Oracle Enterprise Manager Ops Center has functions that are also reached by the cloud infrastructure APIs and CLI. [Table 1-1](#) through [Table 1-11](#) compare the functions provided by the vDC management system with those provided by the cloud infrastructure APIs and CLI.

Table 1-1 vDC Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create, update, and delete vDC	Yes	No
View vDC details	Yes	Yes

Table 1-2 Account Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create, update, and delete account	Yes	No
View account details	Yes	No
Associate or disassociate cloud user with account	Yes	No

Table 1-3 Server Template Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Upload, update, and delete server template	Yes	Yes
Create server template	Yes	No

Table 1–4 vServer Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create, update, and delete vServer	Yes	Yes
Stop, start, and reboot vServer	Yes	Yes
View vServer attributes and metrics	Yes	Yes
Suspend and resume vServer	Yes	No
Shut down all vServers	Yes	No
Send and receive message from vServer	No	Yes
Save vServer as server template	Yes	Yes

Table 1–5 Volume Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create, update, and delete volume	Yes	Yes
Attach volume to and detach volume from vServer	Yes	Yes
Create volume from snapshot	Yes	Yes
Import volume	Yes	Yes

Table 1–6 Snapshot Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create, update, and delete snapshot	Yes	Yes

Table 1–7 Virtual IP Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Allocate and deallocate IP address	Yes	Yes

Table 1–8 Virtual Network Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create, update, and delete private virtual network	Yes	Yes

Table 1–9 Account and vServer Access Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create, import, and delete key pair	Yes	Yes
Create, delete, and access key	Yes	Yes

Table 1–10 Distribution Groups Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create and delete distribution groups	Yes	Yes

Table 1–11 Tag Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Create and delete tags	Yes	Yes

Getting Started

This chapter identifies the prerequisites for the cloud infrastructure APIs and CLI, and describes their installation and configuration.

The following topics are covered in this chapter:

- [Prerequisites](#)
- [Installation](#)
- [Next Steps](#)

Prerequisites

The following prerequisites must be met before using the APIs and CLI:

- Installation of Java Runtime Environment (JRE) or Java Development Kit (JDK) version 1.6 or later appropriate for the platform where the APIs and CLI will be run or installed
- Access to the Enterprise Controller

Installation

This section explains the installation process for the APIs and CLI.

Cloud Infrastructure API

The Web service is automatically installed on the Enterprise Controller as part of the Oracle Enterprise Manager Ops Center installation process. No further steps are required.

Cloud Infrastructure Java Client API

The Java Client API is delivered as a package with the Oracle Enterprise Manager Ops Center product. This package can be installed as a standalone package on a different computer with a connection to the Enterprise Controller.

Package Name	Operating System
ORCL-sysman-iaas-api.pkg	Oracle Solaris
orcl-sysman-iaas-api.rpm	Linux

To install this package, enter the following commands as root user:

- For Oracle Solaris:

```
# cd <repo>/src/dvd/SunOS_i386/Product/components/packages/
# pkgadd -d ORCLsysman-iaas-api.pkg
```

- For Linux:

```
# cd <repo>/src/dvd/Linux_i686/Product/components/packages/
# rpm -i orcl-sysman-iaas-api.rpm
```

The API jar files from the package are stored at */opt/oracle/iaas/iaas-java-api*.

Before you use the API JAR file, set the *JAVA_HOME* environment variable and ensure that the environment variable is part of your *PATH*.

Cloud Infrastructure CLI

The cloud infrastructure CLI is delivered as a separate package with the Oracle Enterprise Manager Ops Center product. This package can be installed as standalone package on a different computer with a connection to the Enterprise Controller.

Package Name	Operating System
ORCLsysman-iaas-cli.pkg	Oracle Solaris
orcl-sysman-iaas-cli.rpm	Linux

To install this package, enter the following commands as root user:

- For Oracle Solaris:

```
# cd <repo>/src/dvd/SunOS_i386/Product/components/packages/
# pkgadd -d ORCLsysman-iaas-cli.pkg
```

- For Linux:

```
# cd <repo>/src/dvd/Linux_i686/Product/components/packages/
# rpm -i orcl-sysman-iaas-cli.rpm
```

The files from the package are stored at */opt/oracle/iaas/cli*. To execute the CLI commands, the user must have permissions to access this directory.

Before using the cloud infrastructure CLI:

1. Set the *JAVA_HOME* environment variable and ensure that the environment variable is part of your *PATH*:

- Korn and bash shells:

```
export JAVA_HOME=<jdk-install-dir>
export PATH=$JAVA_HOME/bin:$PATH
```

- Bourne shell:

```
JAVA_HOME=<jdk-install-dir>
export JAVA_HOME
PATH=$JAVA_HOME/bin:$PATH
export PATH
```

- C shell:

```
setenv JAVA_HOME <jdk-install-dir>
setenv PATH $JAVA_HOME/bin:$PATH
export PATH=$JAVA_HOME/bin:$PATH
```

2. Set the *IAAS_HOME* environment variable:

■ Korn and bash shells:

```
export IAAS_HOME=/opt/oracle/iaas/cli
```

■ Bourne shell:

```
IAAS_HOME=/opt/oracle/iaas/cli  
export IAAS_HOME
```

■ C shell:

```
setenv IAAS_HOME /opt/oracle/iaas/cli
```

Next Steps

Before the cloud administration can use the APIs or the CLI, the cloud administrator must complete the following tasks:

1. Create vDC and an account for the vDC.
2. Create a cloud user and associate the user with the account and vDC.

After getting the cloud user credentials, a cloud user must create an access key for the account in order to manage any resource in the account. For more information, see "[Creating an Access Key](#)".

Managing Resources

This chapter describes how a cloud user can manage different resources in an account by using the APIs and CLI. It also includes a basic example for each action that a cloud user can perform to manage these resources by using the cloud infrastructure API and cloud infrastructure CLI.

For a complete reference and other examples that use the cloud infrastructure API and CLI, see also:

- [Getting Started](#)
- [Cloud Infrastructure API Reference](#)
- [Cloud Infrastructure CLI Reference](#)

Note: For more information about the cloud infrastructure Java client API, see the Javadoc files included in the package of the API.

Overview of Cloud User Resources

This section describes the following different resources that a cloud user can manage by using the APIs and CLI:

- Access keys – Used to manage cloud user access to an account. See "[Managing Account Access](#)" for more information.
- Virtual networks (vNets) – Used for network connectivity of vServers. See "[Managing Virtual Networks](#)" and "[Managing vIP Addresses](#)" for more information.
- Server templates – Used to designate the operating system and how it is installed when creating the vServer. See "[Managing Server Templates](#)" for more information.
- Virtual storage – Includes volumes that can be attached to vServers, and snapshots that capture the current state of a volume for different purposes. See "[Managing Storage](#)".
- Virtual servers (vServers) – Provides the outward interface of a standalone operating system. A vServer has its own identity, local storage, interfaces, and configuration that exist for the full lifetime of the vServer. See "[Managing vServers](#)".

These resources have both attributes and tags:

- Attributes – When a resource is created, it has a set of attributes to describe it. These attributes include defined entities such as name, description, and size. Some

of these attributes can be modified. See ["Managing Attributes of an Account Resource"](#).

- Tags – Used to bind cloud user-specific information to account resources. See ["Managing Tags"](#).

Note: Availability and management of some of these resources might vary depending on the configuration of the vDC. Contact your cloud administrator for more information.

Status of Account Resources

The status attribute is displayed when viewing information or attributes of any of the following account resources:

- vServers
- vNets
- Server templates
- Volumes
- Snapshots
- Distribution groups

This section explains the meaning and usage of the status attribute. The status attribute of an account resource is useful to know because:

- When a cloud user creates a new account resource, the status indicates if the creation job has completed or the account resource creation is still in progress.
- The status field indicates whether an account resource is in a usable state or in an unusable state.

An account resource can have one of the following different states:

- Pending status – Resource creation is in progress.
 - SCHEDULED – Indicates that the creation job is in progress. Wait for the resource to transition from this state before using it.
 - FAILED – Indicates that the creation of the resource failed. A failed job is reflected in the user interface (UI).
- Healthy status – The resource exists. The following values indicate that a resource is usable.
 - OK – Indicates that the resource is healthy and usable as normal.
 - INFO – Indicates that the resource is healthy but an informational event is awaiting attention in the UI.

Note: For vServer resources, the healthy status values of "OK", "INFO", and "UNKNOWN" is swapped with the runtime state of the vServer. Instead of OK or INFO, a vServer can present one of the following status: RUNNING, BLOCKED, PAUSED, SHUTDOWN, DYING, CRASHED, NOSTATE, SUSPENDED, INSTALLING, SHUTDOWNDETACHED, MIGRATING_SRC, and MIGRATING_DST. Of these running vServer states, BLOCKED, DYING, CRASHED, NOSTATE, and FAILED can be regarded as unhealthy.

- Unhealthy status – A resource exists. The following values indicate that a resource might not be usable. This does not apply to the WARNING state.

Note: Which status applies depends on the resource; for example, NEEDS_POWERON does not apply to a volume.

- CRITICAL
- DECONFIGURED
- FAULTED
- MAINTENANCE
- NON_RECOVERABLE
- UNCONFIGURED
- UNINITIALIZED
- UNKNOWN
- NEEDS_POWERON

Operations that delete resources might observe a state transition until the point is reached whereby the resource is no longer visible. For example, the process of terminating a vServer can exhibit a transition to the vServer run state SHUTDOWNDETACHED before the resource is removed.

A delete operation could also fail, in which case the resource must transition to one of the unhealthy states in the preceding list.

Managing Account Access

User management is provided by a central service of Oracle Enterprise Manager Ops Center. The Web service is the entry point for cloud user requests. The Web service uses access keys to authenticate and authorize cloud user requests. A cloud user must create an access key for an account to perform any action for the resources allocated in that account.

An access key consists of an ID, a private key, a public key, and an authentication target account. The private key is used on the client side to sign HTTP requests. The Web service uses the public key to verify incoming HTTP requests and to authenticate the calling user. After creation, the private key is given to the user. The cloud user is responsible for limiting the access to the private key.

The access key management functionality of the Web service is provided by an extra servlet that allows user name and password authentication.

The APIs and CLI provide operations for:

- [Creating an Access Key](#)
- [Viewing Access Key Information](#)
- [Deleting an Access Key](#)

Note: When using the APIs and CLI, an option is available to manage access keys on behalf of another cloud user. This option is available only to cloud administrators.

Creating an Access Key

A cloud user needs an access key to get access to an account and to authenticate all other HTTP requests for an account. Once an access key is created, the public key is stored in the server-side database to verify the incoming HTTP requests and to authenticate the calling user.

When creating an access key by using the CLI, the private key is provided to the user after the access key creation.

If using the RegisterAccessKey by using the API directly, then the cloud user supplies the public key portion of a key pair and manages the private key himself.

Whichever access key creation method is used, a unique access key identifier, such as AK_5, is supplied to the cloud user for direct use of the API. The cloud user is responsible for limiting the access to the private key.

The cloud infrastructure API actions and the cloud infrastructure CLI command required to create an access key are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RegisterAccessKeyRequest	<code>akm-create-access-key</code>
CreateAccessKeyAsObjectRequest	

Example 3-1 Creating an access key using the Cloud Infrastructure API

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=RegisterAccessKey&Version=1&Timestamp=1330975344&Expires=1333975344&account=ACC-0162da5a-5d25-4096-af59-3dd1de27cfad&publicKey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAufVdjdP0MmOLbNypLVMWXfhusawid4Wg4n4FZewSmoBEYA8f8wIA0SI87Shi7RtMcWsEoXvNNHA0wcJoA1RjyVLsI3rtrq0c0k7AxQSwb4UK/rSXW1NXxMh/mE7b3gdA6d9VuwIPnZJ5ZFQUZCLyhaAotLCdACrzbGzYXdqt+rstutT1AVkE2UAMcm503KnIoObZKb8JtepSt74A9RgVBkcCBjmKGfLNOL1K1ZconkITm85TWKRGRFuASxd12ZrD723ZNb66X/a9ebxTMr6vVeskcaZpP1HzvgMOpIyDGwRvxn9yM5WB83zFDGT26LihN/bKzLJXa+F2YNkLrTJQIDAQAB
```

Example 3-2 Creating an access key using the Cloud Infrastructure CLI

```
akm-create-access-key --base-url https://<EnterpriseControllerHostname>/ --user clouser1 --password-file ~/pwd.file --account ACC-4b83b85e-592c-45a1-ba71-3bd0774fbd0e --access-key-file ~/tmp_access_key
```

Viewing Access Key Information

A cloud user can view information about the access keys that the user owns. This information includes a list of the access keys found with the following attributes:

- Access key ID
- Account ID associated with the access key
- Cloud user name associated with the access key

The cloud infrastructure API action and the cloud infrastructure CLI command required to view the access key information are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeAccessKeysRequest	<code>akm-describe-access-keys</code>

Example 3–3 Viewing access key information using the Cloud Infrastructure API

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=DescribeAccessKeys&Version=1&Timestamp=1330954619299&Expires=1330954919299
```

Example 3–4 Viewing access key information using the Cloud Infrastructure CLI

```
akm-describe-access-keys --base-url https://<EnterpriseControllerHostname>/ --user <username> --password-file ~/pwd.file
```

Deleting an Access Key

A cloud user can remove an access key from the system when the access key is not required. The access key deletion deletes and invalidates the registered public key. The cloud infrastructure API action and the cloud infrastructure CLI command required to delete an access key are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DeleteAccessKeyRequest	<code>akm-delete-access-key</code>

Example 3–5 Deleting an access key information using the Cloud Infrastructure API

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=DeleteAccessKey&Version=1&Timestamp=1318278941862&Expires=1318279241862&accessKeyId=AK_3
```

Example 3–6 Deleting an access key information using the Cloud Infrastructure CLI

```
akm-delete-access-key --base-url https://<EnterpriseControllerHostname>/ --user <username> --password-file ~/pwd.file AK_3
```

Viewing vDC Capabilities

A cloud user can view details about the capabilities of the vDC for an account. The vDC capabilities listed include:

- VirtualizationType
- VirtualizationVersion
- ProcessorArchitecture
- ProcessorVersion
- DistributionGroupSupport
- HighAvailabilityDefault
- HighAvailabilityUserControl

The cloud infrastructure API action and the cloud infrastructure CLI command required to view vDC capabilities are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeVdcCapabilitiesRequest	<code>iaas-describe-vdc-capabilities</code>

Example 3–7 Viewing vDC capabilities using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVdcCapabilities&Version=1&Timestamp=1320085185647&Expires=1320085485647&AccessKeyId=AK_3&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
```

Example 3–8 Viewing vDC capabilities using the Cloud Infrastructure CLI

```
./iaas-describe-vdc-capabilities --base-url
https://<EnterpriseControllerHostname>/ -a ak.file -H
```

Managing Virtual Networks

Virtual Network management involves the network connectivity of the vServers in an account. Virtual Network management also includes the connectivity of vServers to an external network.

A vServer is a member of one or more vNets of two different types:

- Private vNet – A private vNet is created based on the private network from the network domain of the virtual datacenter (vDC). Private vNets are only accessible within an account. All vServers that have membership of a private vNet in common can communicate freely through that subnet.
- Public network – This type of networks can be shared among a number of accounts in a vDC. vServers members of public networks also have external communication beyond vDCs and can be used to host public services.

The membership of a vServer in one or more vNets can be specified only at vServer creation time.

The APIs and CLI provide operations for:

- [Creating a vNet](#)
- [Deleting a vNet](#)
- [Viewing vNet Information](#)

Creating a vNet

A cloud user can create private vNets according to the user requirements and within the limits of the account quota.

A public network available in an account is visible to cloud users. However, a cloud user cannot create a public network. The number of public network IP addresses allocated to the cloud user is controlled by the cloud administrator.

The cloud infrastructure API action and the cloud infrastructure CLI command required to create a private vNet are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
CreateVnetRequest	iaas-create-vnet

Example 3–9 Creating a vNet using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=1&AccessKey
Id=AK_
1&name=privatevnetWebApi&Timestamp=1331058639019&Expires=1331058939019&SignatureMe
thod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_
THE_READABILITY_OF_THE_REQUEST
```

Example 3–10 Creating a vNet using the Cloud Infrastructure CLI

```
iaas-create-vnet --base-url https://<EnterpriseControllerHostname>/
--access-key-file ak.file --name myVNET
```

Deleting a vNet

A cloud user can delete only private vNets. The cloud infrastructure API action and the cloud infrastructure CLI command required to delete a private vNet are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DeleteVnetRequest	<code>iaas-delete-vnet</code>

Example 3–11 Deleting a vNet using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteVnet&Version=1&AccessKey
Id=AK_
3&vnet=VNET-00cd848c-771a-4091-b3f4-195a090bbc01&Timestamp=1318283467620&Expires=1
318283767620&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_
HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
```

Example 3–12 Deleting a vNet using the Cloud Infrastructure CLI

```
iaas-delete-vnet --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--vnet VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t
```

Viewing vNet Information

A cloud user can get information about vNets. The cloud user can restrict the vNets included in the result by specifying filters. If there is no match to the given filters, no special message is returned and the response is empty.

When viewing vNets, a list of available vNets for the account is returned with the following attributes:

- ID
- Name
- Description
- Status
- IP address

The cloud infrastructure API action and the cloud infrastructure CLI command required to view vNet information are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeVnetsRequest	<code>iaas-describe-vnets</code>

Example 3–13 Viewing vNet information using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Version=1&Access
KeyId=AK_
3&Timestamp=1318283467620&Expires=1318283767620&SignatureMethod=SHA512withRSA&Sign
atureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST
```

Example 3–14 Viewing vNet information using the Cloud Infrastructure CLI

```
iaas-describe-vnets --base-url https://<EnterpriseControllerHostname>/ -a ak.file
```

Managing vIP Addresses

A virtual IP (vIP) address is a statically configured IP address that is owned by a user.

A vIP address can be allocated from a public network or private vNet. The allocated vIP address is bound to an account and can be associated with a vServer. The vIP address association is performed at vServer creation time. vIP addresses remain bound to vServers even when the vServer is stopped. vIP addresses are not disassociated until the vServer is deleted.

The APIs and CLI provide operations for:

- [Allocating a vIP Address](#)
- [Deallocating a vIP Address](#)
- [Viewing vIP Addresses](#)

Allocating a vIP Address

A cloud user can allocate a vIP address from a public network or a private vNet. When creating a vServer, cloud users can statically assigned the allocated vIP address to the vServer.

The cloud infrastructure API action and the cloud infrastructure CLI command required to allocate a vIP address are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
AllocateIpAddressesRequest	iaas-allocate-ip-addresses

Example 3–15 Allocating a vIP address using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=AllocateIpAddresses&AccessKeyId=AK_32&vnet=VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4&Version=1&Timestamp=1320342206808&Expires=1320342506808&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
```

Example 3–16 Allocating a vIP address using the Cloud Infrastructure CLI

```
iaas-allocate-ip-addresses --base-url https://<EnterpriseControllerHostname>/ -a ak.file --vnet VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c
```

Deallocating a vIP Address

A cloud user can release an allocated vIP address. The vIP address must not be associated with a vServer.

The cloud infrastructure API action and the cloud infrastructure CLI command required to release a vIP address are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
ReleaseIpAddressesRequest	iaas-release-ip-addresses

Example 3–17 Deallocating a vIP address using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=ReleaseIpAddresses&AccessKeyId=AK_32&ipAddresses.2=10.6.0.10&ipAddresses.1=10.6.0.13&vnet=VNET-6ea466f5-6e6b-4159-ad
```

```
f3-8867473d4cf4&Version=1&Timestamp=1320341801846&Expires=1320342101846&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
```

Example 3–18 Deallocating a vIP address using the Cloud Infrastructure CLI

```
iaas-release-ip-addresses --base-url https://<EnterpriseControllerHostname>/ -a
ak.file --vnet VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c --ip-addresses
192.168.0.1
```

Viewing vIP Addresses

A cloud user can get information about the vIP addresses that are allocated and which allocated IP addresses are associated with a vServer in an account. To view unallocated IP addresses associated with vServers, use the actions described in "[Viewing vServer Information](#)".

The cloud user can restrict the vIP addresses included in the result by specifying filters.

The result includes the following details for each vIP address: IP address, vNet ID, and vServer ID.

The cloud infrastructure API action and the cloud infrastructure CLI command required to view allocated vIP address information are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeIpAddressesRequest	<code>iaas-describe-ip-addresses</code>

Example 3–19 Viewing vIP addresses using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeIpAddresses&AccessKeyId=AK_
32&Version=1&Timestamp=1320339663115&Expires=1320339963115&SignatureMethod=SHA512w
ithRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST
```

Example 3–20 Viewing vIP addresses using the Cloud Infrastructure CLI

```
iaas-describe-ip-addresses --base-url https://<EnterpriseControllerHostname>/ -a
ak.file
```

Managing Server Templates

A server template is used to designate the operating system and how it is installed when creating the vServer. Server templates are specific to the processor architecture of the server pool and the virtualization type and might be prebuilt images or identify the operating system distribution.

Server templates are bound to a specific account. Server templates are immutable entities and cannot be changed later, other than changing the name or description.

You require a server template for creating vServers. It has the following attributes:

- Minimum and default vServer type
- User and unique name
- Version
- Owner

- Location
- HA enabled
- Life cycle aware

A new server template can be uploaded as follows:

- Based on an existing vServer
- From a single virtual machine template
- From an assembly. When you upload an assembly, snapshots can be also created as result

The APIs and CLI provide operations for:

- [Uploading a Server Template from a vServer](#)
- [Uploading a Server Template from a URL](#)
- [Uploading a Server Template from an Assembly](#)
- [Deleting a Server Template](#)
- [Viewing Server Template Information](#)

Uploading a Server Template from a vServer

A cloud user can save a server template from a stopped vServer. When a user creates a vServer based on a server template, it is common to install additional software or to configure certain aspects that can be conserved and reused. To conserve the state, the user must save the vServer as a new server template.

The cloud infrastructure API action and the cloud infrastructure CLI command required to upload a server template from a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RegisterServerTemplateFromVserverRequest	<code>iaas-create-server-template-from-vserver</code>

Example 3–21 *Uploading a server template from a vServer using the Cloud Infrastructure API*

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplateFromVserver&AccessKeyId=AK_2&name=mySTfromVserver&vserverId=VSRV-fdba0b48-6e1c-4f41-bde4-2c739dfeeb2&Version=1&Timestamp=1324422880552&Expires=1324423180552f&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
```

Example 3–22 *Uploading a server template from a vServer using the Cloud Infrastructure CLI*

```
iaas-create-server-template-from-vserver --base-url https://<EnterpriseControllerHostname>/ --access-key-file ak.file --name myST --vserver-id VSRV-a959be32-6237-40c7-b44b-47d392e7a0ac
```

Uploading a Server Template from a URL

A cloud user can register a server template based on a file for a single virtual machine template that is ready to be deployed into virtualized platforms. Templates can be of the format .tgz, .tar, or other file types. The file must be accessible through a URL.

The cloud infrastructure API action and the cloud infrastructure CLI command required to register a server template from a URL are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RegisterServerTemplateFromUrlRequest	<code>iaas-create-server-template-from-url</code>

Example 3–23 Uploading a server template from a URL using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplateFromUrl&Version=1&Timestamp=1320096741216&Expires=1320097041216&name=myST&url=http%3A%2F%2Fca-server1.us.oracle.com%2Fvm-templates%2FOVM_EL52_jeos_i386_PVM_WebLogic10gR3_v10.tar.gz&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–24 Uploading a server template from a URL using the Cloud Infrastructure CLI

```
iaas-create-server-template-from-url --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file --name
myAssembly --url http://myServer.com/myTemplate.tar.gz
```

Uploading a Server Template from an Assembly

A cloud user can upload a server template based on an assembly.

An assembly is a collection of interrelated software appliances that can include a configuration of multiple virtual machines with their virtual disks and their interconnectivity. An assembly is contained in a single *.ova* (Open Virtualization Format Archive) file. The assembly file must be accessible through a URL.

When you upload an assembly, snapshots might be also created as result. When an assembly contains multiple virtual machines, each virtual machine is unpacked into its own server template. Each template created has a tag set to indicate which assembly it belongs to. This tag name is *assemblyKey*, and its value is the original assembly ID returned from this action.

The cloud infrastructure API action and the cloud infrastructure CLI command required to register a server template from an assembly are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RegisterServerTemplatesFromAssemblyRequest	<code>iaas-create-server-template-from-assembly</code>

Example 3–25 Uploading a server template from an assembly using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplatesFromAssembly&AccessKeyId=AK_3&name=myST2&url=http%3A%2F%2Fadc4120293.us.oracle.com%3A8888%2FmyAssembly.ova&Version=1&Timestamp=1320097901377&Expires=1320098201377&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
```

Example 3–26 Uploading a server template from an assembly using the Cloud Infrastructure CLI

```
iaas-create-server-template-from-assembly --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file --name
```

```
myAssembly --url http://myServer.com/myAssembly.ova
```

Deleting a Server Template

A cloud user can delete a server template. All data associated with the server template is deleted. The deletion of a server template does not affect any vServers that were created based on the server template.

The cloud infrastructure API action and the cloud infrastructure CLI command required to deregister a server template are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DeregisterServerTemplateRequest	<code>iaas-delete-server-template</code>

Example 3–27 *Deleting a server template using the Cloud Infrastructure API*

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeregisterServerTemplate&AccessKeyId=AK_3&serverTemplateId=TMPL-f089b985-f7fc-4b8a-a5f8-df8f44c95f3c&Version=1&Timestamp=1320098301701&Expires=1320098601701&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
```

Example 3–28 *Deleting a server template using the Cloud Infrastructure CLI*

```
iaas-delete-server-template --base-url https://<EnterpriseControllerHostname>/ --access-key-file ak.file --server-template-id TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeee03
```

Viewing Server Template Information

A cloud user can get information about server templates. The cloud user can restrict the server templates included in the results by specifying filters.

The information includes details about the server templates of an account and all other server templates registered for public use. The server template attributes displayed are:

- ID
- Name
- Description
- Status
- Size
- Public
- Image type
- Read-only

The cloud infrastructure API action and the cloud infrastructure CLI command required to view server template information are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeServerTemplatesRequest	<code>iaas-describe-server-templates</code>

Example 3–29 Viewing server template information using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeServerTemplates&Version=1&Timestamp=1320088801876&Expires=1320089101876&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–30 Viewing server template information using the Cloud Infrastructure CLI

```
iaas-describe-server-templates --base-url https://<EnterpriseControllerHostname>/ --access-key-file ak.file -H
```

Managing Storage

Volumes and snapshots are the two types of storage devices that are managed by a cloud user.

The virtual block storage devices are called volumes. Volumes are serially shareable and can be attached to or detached from a vServer.

A snapshot captures the current state of a volume. Snapshots are immutable.

The APIs and CLI provide operations for:

- [Creating a Volume](#)
- [Deleting a Volume](#)
- [Viewing Volume Information](#)
- [Attaching a Volume to a vServer](#)
- [Detaching a Volume from a vServer](#)
- [Importing a Volume](#)
- [Creating a Snapshot](#)
- [Deleting a Snapshot](#)
- [Viewing Snapshot Information](#)

Creating a Volume

A cloud user can create a shared or non-shared empty volume with a given size. A cloud user can also create a shared or nonshared volume from a snapshot. The new volume can be attached to vServers. Shared volumes can be attached to more than one vServer.

The cloud infrastructure API action and the cloud infrastructure CLI command required to create a volume are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
CreateVolumeRequest	iaas-create-volume

Example 3–31 Creating a volume using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVolume&Version=1&Timestamp=1318462897126&Expires=1318463197126&name=myVol2&size=1&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–32 Creating a volume using the Cloud Infrastructure CLI

```
iaas-create-volume --base-url https://<EnterpriseControllerHostname>/
--access-key-file ak.file --name firstVol --size 2
```

Deleting a Volume

A cloud user can delete a volume. Only volumes that are not currently attached to a vServer can be deleted. The process of deleting a volume does not affect any snapshot that has been created previously based on that volume. The snapshot exists independently of the volume.

The cloud infrastructure API action and the cloud infrastructure CLI command required to delete a volume are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DeleteVolumeRequest	iaas-delete-volume

Example 3–33 Deleting a volume using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteVolume&Version=1&Timesta
mp=1318463692509&Expires=1318463992509&volumeId=VOL-3bbc8f1c-bb4c-4a70-a370-4b6c06
d990dc&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_
OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–34 Deleting a volume using the Cloud Infrastructure CLI

```
iaas-delete-volume --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--volume-id VOL-82b4661e-de12-4931-b9a4-cce8edd23532
```

Viewing Volume Information

A cloud user can get information about the existing volumes in an account. The cloud user can restrict the volumes included in the result by specifying filters. The result includes volumes in all states and has additional information for each volume.

The attributes displayed for each volume are:

- ID
- Name
- Description
- Status
- Shared
- vServers

The cloud infrastructure API action and the cloud infrastructure CLI command required to view details of a volume are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeVolumesRequest	iaas-describe-volumes

Example 3–35 Viewing volume information using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVolumes&Version=1&Time
stamp=1318463287921&Expires=1318463587921&AccessKeyId=AK_2&Signature=SIGNATURE_
HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
```

```
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–36 Viewing volume information using the Cloud Infrastructure CLI

```
iaas-describe-volumes --base-url https://<EnterpriseControllerHostname>/ -a
ak.file -H
```

Attaching a Volume to a vServer

A cloud user can attach volumes to an existing vServer. A cloud user might also attach volumes to a vServer during the vServer creation process. The volume attached is available as a device in the vServer. Depending on the virtualization type of the virtual datacenter, the cloud user might need to stop the vServer before attaching a volume.

The cloud infrastructure API action and the cloud infrastructure CLI command required to attach a volume to a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
AttachVolumesToVserverRequest	<code>iaas-attach-volumes-to-vserver</code>

Example 3–37 Attaching a volume to a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=AttachVolumesToVserver&Version=1&Timestamp=1321384886152&Expires=1321385186152&volumeIds.1=VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–38 Attaching a volume to a vServer using the Cloud Infrastructure CLI

```
iaas-attach-volumes-to-vserver --base-url https://<EnterpriseControllerHostname>/ -a ~/ak.file -vserver-id VSRV-0fb57293-347c-4717-96ef-6dd23154596f --volume-ids VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15
```

Detaching a Volume from a vServer

A cloud user can detach a volume from a vServer. After it is detached, the volume is no longer available as a device in the vServer. Depending on the virtualization type of the vDC, cloud user might need to stop the vServer before detaching a volume.

Caution: Before detaching a volume from a vServer, ensure that the disk being detached is not longer in use to avoid services instability or disk data corruption. Ensure that the vServer properly unmounts the device for the volume; otherwise, data loss or file system corruption might occur

The cloud infrastructure API action and the cloud infrastructure CLI command required to detach a volume from a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DetachVolumesFromVserverRequest	<code>iaas-detach-volumes-from-vserver</code>

Example 3–39 Detaching a volume to a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DetachVolumesFromVserver&Version=1&Timestamp=1321385229679&Expires=1321385529679&volumeIds.1=VOL-052cb4b4-5e56-4
```

```
303-8b3a-82d6ba743a15&force=true&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7
f6&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_
THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–40 Detaching a volume to a vServer using the Cloud Infrastructure CLI

```
iaas-detach-volumes-from-vserver --base-url
https://<EnterpriseControllerHostname>/ -a ~/ak.file -vserver-id
VSRV-0fb57293-347c-4717-96ef-6dd23154596f --volume-ids
VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15 --force
```

Importing a Volume

A cloud user can import a volume from another location. Volumes must be contained in a file of the format *.img*. The volume must be accessible through a URL. The file is uploaded to the vDC, and the volume is created and stored physically. The imported volume is only accessible within the target account.

The cloud infrastructure API action and the cloud infrastructure CLI command required to import a volume are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
ImportVolumeRequest	iaas-import-volume

Example 3–41 Importing a volume using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=ImportVolume&Version=1&Timesta
mp=1324421943593&Expires=1324422243593&shared=true&name=myVolumeImported&url=http%
3A%2F%2Foracle.com%2FES%2FOVM%2Fvolume%2Fvolume1.img&AccessKeyId=AK_
1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–42 Importing a volume using the Cloud Infrastructure CLI

```
iaas-import-volume --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--name myVolume --url http://ovm.oracle.com/volume-image/volume.img
```

Creating a Snapshot

Snapshot creation and management is available if it is supported by the storage of the vDC.

Creating a snapshot serves two main purposes:

- Backup of data stored on a volume
- Creation of new volumes from a snapshot

A cloud user can create a snapshot of a volume or, a snapshot can exist as result of uploading an assembly.

There are four types of exposed disks in an assembly’s template. When snapshots are created from an assembly, three of the disk type are presented to cloud users as snapshots. This allows cloud users to optionally create a shared volume from the snapshot. The three exposed disks are:

- Public Populated – An extra disk with data already present.
- Private Raw – Dynamically created as an empty disk with a size but no disk data.
- Shared Raw – Dynamically created as an empty disk, with a size but no disk data.

The cloud infrastructure API action and the cloud infrastructure CLI command required to create a snapshot of an existing volume are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
CreateSnapshotRequest	iaas-create-snapshot

Example 3–43 Creating a snapshot using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateSnapshot&Version=1&Times
tamp=1318466247111&Expires=1318466547111&volumeId=VOL-246b5c62-4072-41cf-885b-99d6
c63583bd&name=mySnapshot&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_
FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–44 Creating a snapshot using the Cloud Infrastructure CLI

```
iaas-create-snapshot --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--volume-id VOL-246b5c62-4072-41cf-885b-99d6c63583bd --name mySnapshot
```

Deleting a Snapshot

A cloud user can delete a snapshot of a volume. The deletion of a snapshot does not affect any volumes or any other snapshots. Snapshots exist independently of the volume. The cloud infrastructure API action and the cloud infrastructure CLI command required to delete a snapshot are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DeleteSnapshotRequest	iaas-delete-snapshot

Example 3–45 Deleting a snapshot using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteSnapshot&Version=1&Times
tamp=1318624338550&Expires=1318624638550&snapshotId=SNAP-2f2039cd-943b-4072-9ded-e
96b54b7ca79&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–46 Deleting a snapshot using the Cloud Infrastructure CLI

```
iaas-delete-snapshot --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--snapshot-id SNAP-d743e90c-53c5-4b01-a297-e2c944755c48
```

Viewing Snapshot Information

A cloud user can get information about the existing snapshots in an account. The cloud user can restrict the snapshots included in the result by specifying filters. The result includes a list of the snapshots found and additional information for each snapshot.

The attributes displayed for each snapshot are:

- ID
- Name
- Description
- Status

The cloud infrastructure API action and the cloud infrastructure CLI command required to view snapshot information are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeSnapshotsRequest	iaas-describe-snapshots

Example 3–47 Viewing snapshot information using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeSnapshots&Version=1&Timestamp=1318467903784&Expires=1318468203784&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–48 Viewing snapshot information using the Cloud Infrastructure CLI

```
iaas-describe-snapshots --base-url https://<EnterpriseControllerHostname>/ -a ak.file -H
```

Managing Distribution Groups

Distribution groups are necessary for properties similar to anti-affinity scaling. All vServers that are created in a distribution group are placed on separate server of the server pool that the vDC uses. If a spare server is not available, then the create vServer job fails.

The APIs and CLI provide operations for:

- [Creating a Distribution Group](#)
- [Deleting a Distribution Group](#)
- [Viewing Distribution Group Information](#)

Creating a Distribution Group

A cloud user can create distribution groups to associate them with vServer. The membership of a vServer to a distribution group can only be specified at vServer creation time.

When creating a distribution group, a cloud user can specify a maximum number of vServers to be added to the distribution group, the default size is 50000.

The cloud infrastructure API action and the cloud infrastructure CLI command required to create a distribution group are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
CreateDistributionGroupRequest	iaas-create-distribution-group

Example 3–49 Creating a distribution group using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateDistributionGroup&Version=1&Timestamp=1324407098313&Expires=1324407398313&name=myDistributionGroup&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–50 Creating a distribution group using the Cloud Infrastructure CLI

```
iaas-create-distribution-group --base-url https://<EnterpriseControllerHostname>/ -a ak.file --name myDistributionGroup
```

Deleting a Distribution Group

A cloud user can delete distribution groups that are not associated with vServer. The cloud infrastructure API action and the cloud infrastructure CLI command required to delete a distribution group are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DeleteDistributionGroupRequest	<code>iaas-delete-distribution-group</code>

Example 3–51 *Deleting a distribution group using the Cloud Infrastructure API*

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteDistributionGroup&Version=1&Timestamp=1324407269441&Expires=1324407569441&distributionGroupId=DG-8f81381c-a559-4f5b-b45f-086e605a382b&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–52 *Deleting a distribution group using the Cloud Infrastructure CLI*

```
iaas-delete-distribution-group --base-url https://<EnterpriseControllerHostname>/
--access-key-file ak.file --distribution-group-id
DG-3206ed4d-44c9-42c4-8df7-511709a49ead
```

Viewing Distribution Group Information

A cloud user can get information about the existing distribution groups in an account. The cloud user can restrict the distribution groups included in the result by specifying filters. The result includes a list of the distribution groups found and additional information for each distribution group.

The attributes displayed for each distribution group are:

- ID
- Name
- Description
- Status
- vServers
- Size

The cloud infrastructure API action and the cloud infrastructure CLI command required to view details of a distribution group are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeDistributionGroupsRequest	<code>iaas-describe-distribution-groups</code>

Example 3–53 *Viewing distribution groups information using the Cloud Infrastructure API*

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeDistributionGroups&Version=1&Timestamp=1324407152926&Expires=1324407452926&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–54 Viewing distribution groups information using the Cloud Infrastructure CLI

```
iaas-describe-distribution-groups --base-url  
https://<EnterpriseControllerHostname>/ -a ~/ak.file -H
```

Managing vServers

A vServer has its own identity, local storage, interfaces, and configuration that exist for the lifetime of the vServer. vServers are created based on a server template and vServer types. A vServer type defines hardware characteristics of the vServer, such as CPU, memory size, and the network bandwidth.

Each vServer is a member of one or more vNets that define the network connectivity. A vServer has temporary local storage. For hard disk storage, the associated volume is used. A vServer can have one or more volumes attached.

The APIs and CLI provide operations for:

- [Viewing vServer Types](#)
- [Creating a vServer](#)
- [Deleting a vServer](#)
- [Viewing vServer Information](#)
- [Viewing vServer Metrics](#)
- [Stopping a vServer](#)
- [Starting a vServer](#)
- [Rebooting a vServer](#)
- [Sending a Message to a vServer](#)
- [Receiving a Message from a vServer](#)

Viewing vServer Types

A cloud user can get information about all available vServer types for an account. The cloud user can restrict the vServer types included in the result by specifying filters. The result includes a list of the vServer types found and additional information for each vServer type. If the response is an empty list, contact your cloud administrator to verify your cloud user's privileges.

The attributes displayed for each vServer type are:

- ID
- Name
- Description
- Status
- Memory Size
- Storage Size
- vCPU

The cloud infrastructure API action and the cloud infrastructure CLI command required to list the vServer types are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeVserverTypesRequest	<code>iaas-describe-vserver-types</code>

Example 3–55 Listing vServer types using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVserverTypes&Version=1
&Timestamp=1318625453358&Expires=1318625753358&AccessKeyId=AK_
2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST
&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–56 Listing vServer types using the Cloud Infrastructure CLI

```
iaas-describe-vserver-types --base-url https://<EnterpriseControllerHostname>/ -a
ak.file -H
```

Creating a vServer

A cloud user can create a single vServer or multiple vServers at once based on a server template. When a vServer is created, it is automatically started.

When creating a single vServer, the cloud user must use an allocated vIP address. When creating multiple vServer an IP address is automatically assigned from each specified vNets.

When a cloud user creates a vServer, the following resources are required:

- A vServer type
- A server template
- One or more vNets

Depending on the needs of the user and the virtualization type of the vDC, a cloud user can also specify the following resources:

- A key pair
- One or more volumes
- A Distribution group

Additionally to the resources listed, a cloud user might specify the following values when creating a vServer:

- A hostname to override vServer default internal host name, which is same as vServer name.
- A root password
- One or more messages for the vServer
- Enable or disable high availability

High Availability (HA)

When creating a vServer, a cloud user might choose to enable or disable the high availability feature for the vServer. By default, the value of this option is determined by the HighAvailabilityDefault capability of the vDC. If a cloud user chooses not to use the default value, then an option is available to specify a different value; in this case, the HighAvailabilityUserControl capability of the vDC must be enabled. Cloud users can also enable or disable high availability for an existing vServer, see "[Modifying Attributes of an Account Resource](#)".

The cloud infrastructure API action and the cloud infrastructure CLI command required to create and start vServers are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RunVserverRequest or RunVserversRequest	<code>iaas-run-vserver</code> or <code>iaas-run-vservers</code>

Example 3–57 Creating a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=RunVserver&Version=1&Timestamp=1321379615665&Expires=1321379915665&vserverType=457&keyName=myKeyPair&vnets.1=VNET-84ada392-1c13-4f86-8365-1cf7f9c8aadf&name=myVserver&ipAddresses.1=192.168.0.2&serverTemplateId=TMPL-9e4a9ed3-e675-45f1-9d7c-b21c25a55632&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–58 Creating a vServer using the Cloud Infrastructure CLI

```
iaas-run-vserver --base-url https://<EnterpriseControllerHostname>/ -a ~/ak.file --vnets VNET-84ada392-1c13-4f86-8365-1cf7f9c8aadf --key-name myKeyPair --name myVserver --server-template-id TMPL-9e4a9ed3-e675-45f1-9d7c-b21c25a55632 --ip-addresses 192.168.0.2 --vserver-type 457
```

Deleting a vServer

A cloud user can delete a vServer. Deleting a vServer results in shutting down the vServer followed by the deletion of the vServer. The cloud infrastructure API action and the cloud infrastructure CLI command required to terminate a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
TerminateVserversRequest	<code>iaas-terminate-vservers</code>

Example 3–59 Deleting a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=TerminateVservers&Version=1&Timestamp=1321385662260&Expires=1321385962260&vserverIds.1=VSRV-0fb57293-347c-4717-96ef-6dd23154596f&force=true&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–60 Deleting a vServer using the Cloud Infrastructure CLI

```
iaas-terminate-vservers --base-url https://<EnterpriseControllerHostname>/ -a ~/ak.file --vserver-ids VSRV-0fb57293-347c-4717-96ef-6dd23154596f --force
```

Viewing vServer Information

A cloud user can get information about all available vServers for an account. The cloud user can restrict the vServers included in the result by specifying filters. The result includes a list of the vServers found and additional information for each vServer.

The attributes displayed for each vServer are:

- ID – ID of the vServer
- Name – Name of the vServer
- Description – Description of the vServer

- Status – Status of the vServer
- vNets – IDs of the vNets to which the vServer belongs to
- IP addresses – IP addresses associated with the vServer
- Server template ID – ID of the server template that the vServer is based on
- Key pair name – Name of the key pair installed in the vServer for secure shell access
- vServer type – vServer type that the vServer is based on
- HA – Flag to indicate whether the vServer is set to automatic recovery
- Distribution group – ID of the distribution group the vServer belongs to
- Volumes – ID of the volumes associated with the vServer
- vCPU – Number of virtual CPUs allocated to a vServer
- Memory – Total memory allocated to a vServer
- Dedicated storage – Total storage dedicated to the vServer
- Attached storage – Total storage of the volumes attached to the vServer
- Tags – List of tags associated with the vServer

The cloud infrastructure API action and the cloud infrastructure CLI command required to view vServer information are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeVserversRequest	iaas-describe-vservers

Example 3–61 Viewing vServer information using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVservers&Version=1&Timestamp=1320105338731&Expires=1320105638731&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–62 Viewing vServer information using the Cloud Infrastructure CLI

```
iaas-describe-vservers --base-url https://<EnterpriseControllerHostname>/ -a ak.file -H
```

Viewing vServer Metrics

A cloud user can get information about the cpu usage of the vServers. The cloud user can restrict the vServers included in the result by specifying filters. The result includes a list of the vServers found and additional information for each vServer.

The attributes displayed for each vServer are:

- ID – ID of the vServer
- Name – Name of the vServer
- Description – Description of the vServer
- Status – Status of the vServer
- CPU usage – CPU usage of the vServer
- Tags – List of tags associated with the vServer

The cloud infrastructure API action and the cloud infrastructure CLI command required to view vServer metrics are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeVserverMetricsRequest	iaas-describe-vserver-metrics

Example 3–63 Viewing vServer metrics using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVserverMetrics&Version=1&Timestamp=1320105338731&Expires=1320105638731&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–64 Viewing vServer metrics using the Cloud Infrastructure CLI

```
iaas-describe-vserver-metrics --base-url https://<EnterpriseControllerHostname>/ -a ak.file -H
```

Stopping a vServer

A cloud user can stop a vServer. When a vServer is stopped, the guest operating system is shut down. A stopped vServer can be restarted later. The cloud infrastructure API action and the cloud infrastructure CLI command required to stop a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
StopVserversRequest	iaas-stop-vservers

Example 3–65 Stopping a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=StopVservers&Version=1&Timestamp=1320105610783&Expires=1320105910783&vserverIds.1=VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–66 Stopping a vServer using the Cloud Infrastructure CLI

```
iaas-stop-vservers --base-url https://<EnterpriseControllerHostname>/ -a ak.file --vserver-ids VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291
```

Starting a vServer

A cloud user can start a vServer that has been stopped. The restarted vServer might not get the same IP address as the original vServer. The cloud infrastructure API action and the cloud infrastructure CLI command required to start a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
StartVserversRequest	iaas-start-vservers

Example 3–67 Starting a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=StartVservers&Version=1&Timestamp=1320104759496&Expires=1320105059496&vserverIds.1=VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–68 Starting a vServer using the Cloud Infrastructure CLI

```
iaas-start-vservers --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--vserver-ids VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291
```

Rebooting a vServer

A cloud user can reboot a vServer. Rebooting a vServer does not change the IP address and other resources of the vServer. The cloud infrastructure API action and the cloud infrastructure CLI command required to reboot a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RebootVserversRequest	iaas-reboot-vservers

Example 3–69 Rebooting a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=RebootVservers&Version=1&Times
tamp=1321380470326&Expires=1321380770326&vserverIds.1=VSRV-0fb57293-347c-4717-96ef
-6dd23154596f&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–70 Rebooting a vServer using the Cloud Infrastructure CLI

```
iaas-reboot-vservers --base-url https://<EnterpriseControllerHostname>/ -a
~/ak.file --vserver-ids VSRV-0fb57293-347c-4717-96ef-6dd23154596f
```

Sending a Message to a vServer

A cloud user can send messages to a vServer and these messages can be read from the guest operating system. The messages are sent to a common buffer in the virtualization layer that has a limited size. This action might return an error if the size limit is reached or if the vServer is not in appropriate state to send messages.

The cloud infrastructure API action and the cloud infrastructure CLI command required to send messages to a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
SendMessageToVserverRequest	iaas-send-messages-to-vserver

Example 3–71 Sending messages to a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=SendMessageToVserver&Version=1
&Timestamp=1321381585564&Expires=1321381885564&messages.1.value=Running&messages.1
.key=myStatus&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&AccessKeyId=AK_
1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–72 Sending messages to a vServer using the Cloud Infrastructure CLI

```
iaas-send-messages-to-vserver --base-url https://<EnterpriseControllerHostname>/
-a ~/ak.file --vserver-id VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6 --messages
com.oracle.ovab.test=Running
```

Receiving a Message from a vServer

A cloud user can read a message sent by a vServer. The message is read from the common buffer in the virtualization layer by name. This action might return an error if the vServer is not in appropriate state to receive messages.

The cloud infrastructure API action and the cloud infrastructure CLI command required to receive a message from a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
ReceiveMessageFromVserverRequest	iaas-receive-message-from-vserver

Example 3–73 Receiving message from a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=ReceiveMessageFromVserver&Version=1&Timestamp=1321381585564&Expires=1321381885564&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&key=myMessageKey&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–74 Receiving message from a vServer using the Cloud Infrastructure CLI

```
iaas-receive-message-from-server --base-url
https://<EnterpriseControllerHostname>/ -a ~/ak.file --vserver-id
VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6 --key com.oracle.ovab.test-response
```

Managing Key Pairs

The Web service provides the key pair management functionality. Key pair management defines a way to get the credentials for user authentication to access the guest operating system. Key pairs restrict access to vServers created by a cloud user based on common server templates. The credentials to supply are chosen by the cloud user before the vServer is created or run for the first time. Key pairs are managed per account and they are only visible as tags.

The APIs and CLI provide operations for:

- [Creating a Key Pair](#)
- [Importing a Key Pair](#)
- [Deleting a Key Pair](#)
- [Viewing Key Pair Information](#)

Creating a Key Pair

A cloud user can create a key pair to be associated with a vServer as part of the create vServer operation. When a key pair is created, a public key is stored in the application and a private key is handed out to the user.

The key pair is typically an RSA 2048-bit key. If a different key pair strength is required, use the Import Key Pair action.

The cloud infrastructure API action and the cloud infrastructure CLI command required to create a key pair are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RegisterKeyPairRequest	iaas-create-key-pair

Example 3–75 Creating a key pair using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterKeyPair&Version=1&Timestamp=1318279380444&Expires=1318279680444&publicKey=ssh-rsa+AAAAB3NzaC1yc2EAAAADAQABAAQDg2eienGE4vEMSMcVMbYbC8z2q%2FvHz3H6AanlJ6B4udseK%0A8CpaHJ23eGwCjcgAmuZCJ%2
```

```
FOoHUA2dN2PNPuK6g%2BZndR8wVaaQT89eWDZx9oaf0%2F2Eg%2FLeKJ3moVH%0AvIYvFB9aFCpa4H%2BO
mLfm%2FmfQ4CYeDfo0r0jxCCB0YLo0876LQqK5X%2BtgRXwbAbPH2Mzbzp%2FzdkQ%0ArsBqSgUQ%2B1V
4LkN6TQe06P5a2QYI1UhRXwUorTnbXczGq9zEJJ7ef%2F74xIQZfAipkYkyGgktsXrM%0A%2F%2Bs789v9
ipaDB5B26y3aqjIdvW4ZLDvuGXPs60aiUfj2WGIqx0KSVL%2FyB%2FtK1WbuZYw1+IaaS-Generated&ke
yName=myKeyPairK&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–76 Creating a key pair using the Cloud Infrastructure CLI

```
iaas-create-key-pair --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--key-name myKeyPair --key-file myKeyFile
```

Importing a Key Pair

A cloud user can import a public key pair that was created with a third-party tool using the RSA encryption algorithm. The length of the formatted key string is limited to 2048 characters.

When importing a key pair, the cloud user supplies the file that stores the public key and then the public key is registered. The cloud user manages the private key himself.

The cloud infrastructure API action and the cloud infrastructure CLI command required to import a key pair are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
ImportKeyPairRequest , RegisterKeyPairRequest	iaas-import-key-pair

Example 3–77 Importing a key pair using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterKeyPair&Version=1&Time
stamp=1324421510757&Expires=1324421810757&publicKey=ssh-rsa+AAAAB3NzaC1yc2EAAAABIw
AAQEAtJnFD8INGltM%2FQI1xkdjh4t2R4%2FtTmPUDzMIRP%0A&keyName=myKeyPair&AccessKeyId=
AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–78 Importing a key pair using the Cloud Infrastructure CLI

```
iaas-import-key-pair --base-url https://<EnterpriseControllerHostname>/ -a ak.file
--key-name myOtherPair --key-file myOtherPair.key
```

Deleting a Key Pair

A cloud user can delete a key pair from an account. When a key pair is deleted, the public key registered in the application is deleted and no longer valid.

The cloud infrastructure API action and the cloud infrastructure CLI command required to delete a key pair are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DeleteKeyPairRequest	iaas-delete-key-pair

Example 3–79 Deleting a key pair using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteKeyPair&Version=1&Timest
amp=1318281994430&Expires=1318282294430&keyName=myKeyPair&AccessKeyId=AK_
3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–80 Deleting a key pair using the Cloud Infrastructure CLI

```
iaas-delete-key-pair --base-url https://<EnterpriseControllerHostname>/
--access-key-file ak.file --key-name myKeyPairName
```

Viewing Key Pair Information

A cloud user can get information about the existing key pairs in an account. The cloud user can restrict the key pairs included in the result by specifying filters. The result includes the name and fingerprint for each key pair found.

The cloud infrastructure API action and the cloud infrastructure CLI command required to view information about a key pair are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeKeyPairsRequest	<code>iaas-describe-key-pairs</code>

Example 3–81 Viewing key pair information using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeKeyPairs&Version=1&Tim
estamp=1320361610130&Expires=1320361910130&AccessKeyId=AK_32&Signature=
SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–82 Viewing key pair information using the Cloud Infrastructure CLI

```
iaas-describe-key-pairs --base-url https://<EnterpriseControllerHostname>/ -a
ak.file
```

Managing Attributes of an Account Resource

A set of attributes exists for each type of account resource in an account. The management of attributes of an account resource includes viewing and modifying the value of specific attributes for an account resource. A cloud user can only manage the attributes of the following account resources: volumes, snapshots, vServers, server templates, and vNets.

The APIs and CLI provide operations for:

- [Viewing Attributes of an Account Resource](#)
- [Modifying Attributes of an Account Resource](#)

Viewing Attributes of an Account Resource

A cloud user can view the value of specific attributes of an account resource. To view the attribute value, the cloud user needs to specify:

- ID of a valid account resource.
- One or more attribute names for the account resource. Attribute names for a resource are listed when viewing an account resource information.

The cloud infrastructure API action and the cloud infrastructure CLI command required to view the attributes of an account resource are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeAttributesRequest	<code>iaas-describe-attributes</code>

Example 3–83 Viewing attributes of an account resource using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeAttributes&Version=1&Timestamp=1318464041596&Expires=1318464341596&attrNames.4=shared&resourceId=VOL-246b5c62-4072-41cf-885b-99d6c63583bd&attrNames.3=size&attrNames.2=status&attrNames.1=id&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–84 Viewing attributes of an account resource using the Cloud Infrastructure CLI

```
iaas-describe-attributes --base-url https://<EnterpriseControllerHostname>/ -a ak.file --id VNET-7403e87f-1bab-4097-98ae-ea72d8fe4b3f --attr-names name
```

Modifying Attributes of an Account Resource

A cloud user can modify the values of the attributes of the following account resources: volumes, snapshots, vServers, server templates, and vNets. The attributes that are editable for these account resources are name and description.

When the HighAvailabilityUserControl capability is enabled in the vDC, cloud users can also enable or disable high availability for an existing vServer by modifying the HA attribute of the vServer.

The cloud infrastructure API action and the cloud infrastructure CLI command required to modify the attributes of a resource are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
ModifyAttributesRequest	iaas-modify-attributes

Example 3–85 Modifying vNet attributes using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=ModifyAttributes&Version=1&Timestamp=1318464546890&Expires=1318464846890&resourceId=VOL-246b5c62-4072-41cf-885b-99d6c63583bd&attributes.2.value=myNewDescription&attributes.2.name=description&attributes.1.name=name&attributes.1.value=myNewName&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–86 Modifying vNet attributes using the Cloud Infrastructure CLI

```
iaas-modify-attributes --base-url https://<EnterpriseControllerHostname>/ -a ak.file --id VNET-7403e87f-1bab-4097-98ae-ea72d8fe4b3f --attributes name=myNewName
```

Managing Tags

A tag is a key and value pair that can be attached to a resource. The key and the value are strings. All entities in Oracle Enterprise Manager Ops Center are managed resources that can be tagged. Tags are used to bind user-specific information to entities.

The APIs and CLI provide operations for:

- [Creating a Tag](#)
- [Removing a Tag](#)
- [Viewing Tag Information](#)

Creating a Tag

A cloud user can create a tag. Creating a tag adds a new tag or overwrites an existing tag. The cloud infrastructure API action and the cloud infrastructure CLI command required to create a tag are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
CreateTagsRequest	<code>iaas-create-tags</code>

Example 3–87 *Creating a tag using the Cloud Infrastructure API*

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateTags&Version=1&Timestamp=1320101484395&Expires=1320101784395&resourceId=VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50&tags.1.name=myTag&tags.1.value=myTagValue&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–88 *Creating a tag using the Cloud Infrastructure CLI*

```
iaas-create-tags --base-url https://<EnterpriseControllerHostname>/ -a ak.file --id VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t --tags myTag=myTagValue
```

Removing a Tag

A cloud user can remove tags from an account resource of an account. The cloud infrastructure API action and the cloud infrastructure CLI command required to remove a tag are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeTagsRequest	<code>iaas-delete-tags</code>

Example 3–89 *Deleting a tag using the Cloud Infrastructure API*

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteTags&Version=1&Timestamp=1320102752960&Expires=1320103052960&resourceId=VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50&tags.1.name=myTag&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–90 *Deleting a tag using the Cloud Infrastructure CLI*

```
iaas-delete-tags --base-url https://<EnterpriseControllerHostname>/ -a ak.file --id VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t --tags myTag
```

Viewing Tag Information

A cloud user can get information about tags attached to an account resource. The cloud user can restrict the tags included in the result by specifying filters. The result includes the ID of the account resource, along with the name and value of the tags found. The cloud infrastructure API action and the cloud infrastructure CLI command required to view tags information are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeTagsRequest	<code>iaas-describe-tags</code>

Example 3–91 Viewing tag information using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeTags&Version=1&Timesta  
mp=1320101837081&Expires=1320102137081&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_  
BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_  
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–92 Viewing tag information using the Cloud Infrastructure CLI

```
iaas-describe-tags --base-url https://<EnterpriseControllerHostname>/ -a ak.file
```

Cloud Infrastructure API Reference

This chapter describes some characteristics of the cloud infrastructure API. It also presents a description of the information contained in a typical request to the cloud infrastructure API, and contains all actions and data types information.

The following topics are covered in this chapter:

- [Overview of the Cloud Infrastructure API](#)
- [Cloud Infrastructure API Requests](#)
- [List of Actions](#)
- [Data Types](#)

Overview of the Cloud Infrastructure API

The cloud infrastructure API is a Web service API implemented in Oracle Enterprise Manager Ops Center that exposes its virtual datacenter management system. This Web service provides access to a subset of the virtual datacenter functionality available through the Oracle Enterprise Manager Ops Center user interface.

The functionality exposed by the Web service can then be accessed programmatically using the Java client API, the CLI, or by making calls directly to the Web service.

This functionality includes the management of vServers, server templates, network resources, and storage resources for an account.

Cloud Infrastructure API Requests

The Web service was developed as a simple query Web service. This Web service requires an SSL encrypted HTTP connection. A typical request to the Web service is made through HTTP using a GET or POST method that includes the request parameters for the desired action. The Web service returns the result of a request in XML format that conforms to the XML schema of the Web service.

The Web service implementation added the following servlets deployed into the Common Agent Container of Oracle Enterprise Manager Ops Center:

- The AKM servlet – The AKM servlet handles all actions related to account authentication.
- The IAAS servlet – The IAAS servlet handles all actions related to cloud computing management.

Required Web Service HTTP Request Parameters

All requests to the Web service share some required parameters. Other parameters are only required for requests to the specific servlet, AKM or IAAS. [Table 4–1](#) contains the common parameters that are required in an HTTP request to the Web service.

Table 4–1 Required Parameters for an HTTP Request

Servlet	Parameter	Description
AKM and IAAS	Action	Indicates the name of the IAAS or AKM action to perform
AKM and IAAS	Version	Version of the Web service
IAAS only	AccessKeyId	Access Key ID. This parameter identifies the user and the account. This is the value returned by the <i>RegisterAccessKey</i> action.
AKM and IAAS	Timestamp	Timestamp in milliseconds since January 1, 1970
AKM and IAAS	Expires	Time at which the request expires. Expiry date in milliseconds since January 1, 1970; it must be greater than the timestamp.
IAAS only	Signature	Encoded hash value of the data to be signed. Signature is used to authenticate the data.
IAAS only	SignatureMethod	Type of the signature
IAAS only	SignatureVersion	Version number of the signature

Signing a Web Service Request

Signing a request to the Web service involves applying the following algorithm.

1. Collect all parameters for the Web service request (see [Table 4–1](#)). The parameters and their values are UTF-8 strings. The signature is not yet included in these parameters because the aim of the algorithm is to calculate the value of this signature parameter.
2. The URL encodes all parameter names and their values. The following characters are not URL encoded and remain as they are: A to Z, a to z, 0 to 9, minus (-), underscore (_), period (.), and asterisk (*). All other characters are encoded as %XX (with XX being a hexadecimal number consisting of 0 to 9 and/or uppercase A to F). Extended UTF-8 characters are encoded with %XX%XX. Space is encoded as a plus sign (+). This is what Java's `java.net.URLEncoder.encode()` method does, using UTF-8 as encoding.
3. Create a string by concatenating the next elements in the same order. A "\n" is required between every field.
 - a. POST or GET, depending on request type. "\n"
 - b. Host name as given in the HTTP host header field, all in lower case (for example, `opscenter.com`) "\n"
 - c. Base URL of the Web service. "\n"
 - d. A sequence of the sorted, encoded parameter names, followed directly by the equal sign (=) and their corresponding encoded parameter value. The parameters are joined together with the ampersand (&). "\n"

Sign this string using the private key and the specified `SignatureMethod` and `SignatureVersion` parameters. The signature method must be `SHA512withRSA`.

The signing result is then used as the value of the signature parameter for the request.

[Example 4-1](#) and [Example 4-2](#) show a complete URL for an HTTP GET request to the IAAS and AKM servlet. For a detailed example of generating requests to the Web service, see [Appendix A](#).

Example 4-1 URL for an HTTP GET request to the IAAS servlet:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Version=1&Access  
KeyId=AK_  
1&Timestamp=1331058169938&Expires=1331058469938&SignatureMethod=SHA512withRSA&Sign  
atureVersion=1&Signature=bj8GfJCqvPZZPU2JoWAGzZdCF+N767rQejILMQwNdgKLfoGGqAwDPRYMr  
/ghUoBc6RB3nKYgAyPdmtCfhzRGTqECgUWy0jCrE99+utGeeJ0/XRQ9LxyYeBgZj03lHP+hFhUo+gUtQaS  
YPhUHH7eTkxg/CrolMxibglypJM/rIf90yEqSeqhphQt7hWxlT0DNAY6/cZt8isT/Tu8V7ZFjBFkEpLfn9  
7bIOJ2vIIPoetmftuw4ObtqjbUp6+7dpVkhkCQnX0MAIDj+mjoreOzcwK+F1pYuzES0fjaW0MowG+ca/9  
gttDjg7r5H29i3qbbjI1vAt6fk1HPpSxQTSTOTg==
```

Example 4-2 URL for an HTTP GET request to the AKM servlet:

```
https://<<username>>:<<password>>@<EnterpriseControllerHostname>/akm/?Action=  
DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1330954919299
```

List of Actions

The following actions are described in this section:

Functionality	Actions
Account access	CreateAccessKeyAsObjectRequest, CreateAccessKeyToFileRequest, DeleteAccessKeyRequest, DescribeAccessKeysRequest, DescribeAccountsRequest, RegisterAccessKeyRequest
Server template management	DeregisterServerTemplateRequest, DescribeServerTemplatesRequest, RegisterServerTemplateFromUrlRequest, RegisterServerTemplateFromVserverRequest, RegisterServerTemplatesFromAssemblyRequest
Virtual network management	CreateVnetRequest, DeleteVnetRequest, DescribeVnetsRequest
vIP address management	AllocateIpAddressesRequest, DescribeIpAddressesRequest, ReleaseIpAddressesRequest
Volume management	AttachVolumesToVserverRequest, CreateVolumeRequest, DeleteVolumeRequest, DescribeVolumesRequest, DetachVolumesFromVserverRequest, ImportVolumeRequest
Snapshot management	CreateSnapshotRequest, DeleteSnapshotRequest, DescribeSnapshotsRequest
Key pair management	CreateKeyPairAsObjectRequest, CreateKeyPairToFileRequest, DeleteKeyPairRequest, DescribeKeyPairsRequest, ImportKeyPairRequest
vServer management	DescribeVserversRequest, DescribeVserverTypesRequest, RebootVserversRequest, ReceiveMessageFromVserverRequest, RunVserverRequest, RunVserversRequest, SendMessageToVserverRequest, StartVserversRequest, StopVserversRequest, TerminateVserversRequest
Distribution group management	CreateDistributionGroupRequest, DeleteDistributionGroupRequest, DescribeDistributionGroupsRequest
Resource attribute management	DescribeAttributesRequest, ModifyAttributesRequest
Tag management	CreateTagsRequest, DeleteTagsRequest, DescribeTagsRequest
vDC capabilities	DescribeVdcCapabilitiesRequest

AllocateIpAddressesRequest

Description

Action to allocate a number of IP addresses from a vNET.

Request Parameters

vnet

ID of the vNET.

Type: *VnetIdType*

Default: *None*

Required: *Yes*

num

Number of IP addresses to reserve.

Type: *PositiveInteger*

Default: *1*

Required: *No*

Result Elements

The result elements are contained in `AllocateIpAddressesResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

ipAddresses

List of reserved IP addresses.

Type: *xs:string*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example allocates an IP address from a vNET.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=AllocateIpAddresses&Version=1&Timestamp=1320342206808&Expires=1320342506808&vnet=VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4&AccessKeyId=AK_32&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="AllocateIpAddressesResult"  
requestId="403"><ipAddresses>10.6.0.13</ipAddresses></result>
```

Example 2

The following example allocates two IP addresses from the specified vNET.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=AllocateIpAddresses&Version=1&  
Timestamp=1320341343286&Expires=1320341643286&num=2&vnet=VNET-6ea466f5-6e6b-4159-a  
df3-8867473d4cf4&AccessKeyId=AK_32&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_  
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result  
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="AllocateIpAddressesResult"  
requestId="399"><ipAddresses>10.6.0.1</ipAddresses><ipAddresses>10.6.0.2</ipAdres  
ses></result>
```

See Also

- [DeleteVnetRequest](#)
- [DescribeIpAddressesRequest](#)
- [DescribeVnetsRequest](#)
- [ReleaseIpAddressesRequest](#)

AttachVolumesToVserverRequest

Description

Action to attach one or more volumes to a vServer.

Request Parameters

vserverId

ID of the vServer.

Type: *VserverIdType*

Default: *None*

Required: *Yes*

volumeIds

List of volume IDs.

Type: *VolumeIdType*

Default: *None*

Required: *Yes*

Minimum: *1*

Maximum: *Unbounded*

Result Elements

The result elements are contained in `AttachVolumesToVserverResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example attaches a volume to the specified vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=AttachVolumesToVserver&Version=1&Timestamp=1321384886152&Expires=1321385186152&volumeIds.1=VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AttachVolumesToVserverResult" requestId="151"/>
```

See Also

- [CreateVolumeRequest](#)
- [DescribeVolumesRequest](#)
- [DeleteVolumeRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [DescribeAttributesRequest](#)
- [ImportVolumeRequest](#)
- [ModifyAttributesRequest](#)

CreateAccessKeyAsObjectRequest

Description

Action to create an access key for a cloud user for the specified account. This action also allows the creation of an access key on behalf of another user.

Request Parameters

account

ID of the account.

Type: *AccountIdType*

Default: *None*

Required: *Yes*

forUser

User name of a different cloud user. A user can create an access key on behalf of another cloud user. This option requires cloud administrator privileges.

Type: *GenericString*

Default: *None*

Required: *No*

Result Elements

The `CreateAccessKeyAsObjectResult` type is created by hand as it contains the `AccessKey` class and is only required on the client side.

Examples

No examples are provided for this action. See the `RegisterAccessKeyRequest` action for related examples.

See Also

- [CreateAccessKeyToFileRequest](#)
- [DeleteAccessKeyRequest](#)
- [DescribeAccessKeysRequest](#)
- [DescribeAccountsRequest](#)
- [RegisterAccessKeyRequest](#)

CreateAccessKeyToFileRequest

Description

Action to create the access key file of a cloud user for the specified account. This action also allows the creation of an access key on behalf of another user.

Request Parameters

account

ID of the account.

Type: *AccountIdType*

Default: *None*

Required: *Yes*

keyStoreFileName

Name of a file to store the private key of the access key.

Type: *GenericString*

Default: *None*

Required: *Yes*

forUser

User name of a different cloud user. User using this option must have cloud administrator privileges.

Type: *GenericString*

Default: *None*

Required: *No*

Result Elements

The CreateAccessKeyToFileResult structure contains:

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

accessKeyId

ID of the access key.

Type: *xs:string*

Examples

No examples are provided for this action. See the RegisterAccessKeyRequest action for related examples.

See Also

- [CreateAccessKeyAsObjectRequest](#)
- [DeleteAccessKeyRequest](#)

- [DescribeAccessKeysRequest](#)
- [DescribeAccountsRequest](#)
- [RegisterAccessKeyRequest](#)

CreateDistributionGroupRequest

Description

Action to create a distribution group for an account.

Request Parameters

name

Name of the distribution group.

Type: *GenericString*

Required: *Yes*

description

Description of the distribution group.

Type: *GenericString*

Required: *No*

size

Size of the distribution group. The size is the maximum number of vServers that can be added to the distribution group.

Type: *PositiveInteger*

Required: *No*

Default: *50000*

Result Elements

The CreateDistributionGroupResult structure contains:

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

distributionGroupId

ID of the distribution group.

Type: *xs:string*

Examples

Example 1

The following example creates a distribution group for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateDistributionGroup&Version=1&Timestamp=1324407098313&Expires=1324407398313&name=myDistributionGroup&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CreateDistributionGroupResult"
requestId="103"><distributionGroupId>DG-8f81381c-a559-4f5b-b45f-086e605a382b</dist
ributionGroupId></result>
```

Example 2

The following example creates a distribution group with a size of 10.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateDistributionGroup&Versio
n=1&Timestamp=1324407098313&Expires=1324407398313&name=myDistributionGroup2&size=1
0&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_
THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CreateDistributionGroupResult"
requestId="103"><distributionGroupId>DG-9eb288c2-85e7-4392-80a6-d1c1709de4cd</dist
ributionGroupId></result>
```

See Also

- [DeleteDistributionGroupRequest](#)
- [DescribeDistributionGroupsRequest](#)

CreateKeyPairAsObjectRequest

Description

Action to create a key pair. This is a client-side action only, it does not travel through the network.

Request Parameters

keyName

Name of the key pair to create.

Type: *xs:string*

Default: *None*

Required: *Yes*

Result Elements

The CreateKeyPairAsObjectResult type is created by hand as it contains the KeyPair class and is only needed on the client side.

Examples

No examples are provided for this action. See the RegisterKeyPairRequest action for related examples.

See Also

- [CreateKeyPairToFileRequest](#)
- [DeleteKeyPairRequest](#)
- [DescribeKeyPairsRequest](#)
- [ImportKeyPairRequest](#)
- [RegisterKeyPairRequest](#)

CreateKeyPairToFileRequest

Description

Action to create the key pair file for an account.

Request Parameters

keyName

Name of the key pair to create.

Type: *GenericString*

Default: *None*

Required: *Yes*

keyFileName

Name of the file to store the private key of the key pair.

Type: *GenericString*

Default: *None*

Required: *Yes*

Result Elements

The CreateKeyPairToFileResult structure contains:

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

keyName

Name of the key pair.

Type: *xs:string*

keyFingerprint

Key pair fingerprint.

Type: *xs:string*

Examples

No examples are provided for this action. See the RegisterKeyPairRequest action for related examples.

See Also

- [CreateKeyPairAsObjectRequest](#)
- [DeleteKeyPairRequest](#)
- [DescribeKeyPairsRequest](#)
- [ImportKeyPairRequest](#)
- [RegisterKeyPairRequest](#)

CreateSnapshotRequest

Description

Action to create a snapshot for an account based on an existing volume.

Request Parameters

name

Name of the snapshot.

Type: *GenericString*

Required: *Yes*

description

Description of the snapshot.

Type: *GenericString*

Required: *No*

volumeId

ID of the volume.

Type: *VolumeIdType*

Required: *Yes*

Result Elements

The result elements are contained in *CreateSnapshotResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

snapshotId

ID of the snapshot.

Type: *SnapshotIdType*

Examples

Example 1

The following example creates a snapshot based on the specified volume ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateSnapshot&Version=1&Times
tamp=1318466247111&Expires=1318466547111&volumeId=VOL-246b5c62-4072-41cf-885b-99d6
c63583bd&name=mySnapshot&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_
FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
```

```
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="CreateSnapshotResult"  
requestId="118"><snapshotId>SNAP-d743e90c-53c5-4b01-a297-e2c944755c48</snapshotId>  
</result>
```

See Also

- [CreateVolumeRequest](#)
- [DeleteSnapshotRequest](#)
- [DeleteVolumeRequest](#)
- [DescribeSnapshotsRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

CreateTagsRequest

Description

Action to create or overwrite one or more tags for an existing resource.

Request Parameters

resourceId

ID of the resource.

Type: *ResourceIdType*

Default: *None*

Required: *Yes*

tags

List of one or more tag names and values.

Type: *TagType*

Default: *None*

Required: *Yes*

Minimum: *1*

Maximum: *Unbounded*

Result Elements

The result elements are contained in *CreateTagsResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example creates a tag for the specified virtual network ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateTags&Version=1&Timestamp=1320101484395&Expires=1320101784395&resourceId=VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50&tags.1.name=myTag&tags.1.value=myTagValue&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CreateTagsResult" requestId="269"/>
```

See Also

- [DeleteTagsRequest](#)
- [DescribeTagsRequest](#)

CreateVnetRequest

Description

Action to create a virtual network for an account.

Request Parameters

name

Name of the virtual network.

Type: *GenericString*

Default: *None*

Required: *Yes*

description

A description for the virtual network.

Type: *GenericString*

Default: *None*

Required: *No*

size

Size of the virtual network's subnet.

Type: *PositiveInteger*

Default: *16*

Required: *No*

Result Elements

The result elements are contained in `CreateVnetResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

vnetId

ID of the virtual network.

Type: *GenericString*

Examples

Example 1

The following example creates a virtual network for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=1&Timestamp=1318282423837&Expires=1318282723837&name=myVnet&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CreateVnetResult"
requestId="140"><vnetId>VNET-0a1b00f5-b87e-4dcc-9047-9d396a44b4d5</vnetId></result
>
```

Example 2

The following example creates a virtual network for an account and adds a description of it.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=1&Timestamp
=1318282726109&Expires=1318283026109&description=DescriptionofmyVnet2&name=myVnet2
&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CreateVnetResult"
requestId="141"><vnetId>VNET-00cd848c-771a-4091-b3f4-195a090bbc01</vnetId></result
>
```

See Also

- [DeleteVnetRequest](#)
- [DescribeVnetsRequest](#)
- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

CreateVolumeRequest

Description

Action to create a volume for an account.

Request Parameters

name

Name of the volume.

Type: *GenericString*

Default: *None*

Required: *Yes*

description

Description of the volume.

Type: *GenericString*

Default: *None*

Required: *No*

size

Size of the volume in gigabytes (GB). This option should not be used if a snapshot ID is given. An error is generated if both `size` and `snapshotId` parameters are used.

Type: *PositiveInteger*

Default: *16*

Required: *No*

shared

Flag to indicate if the volume is shared with other accounts in a vDC.

Type: *xs:boolean*

Default: *0 (false)*

Required: *No*

snapshotId

ID of a snapshot. This option should not be used if a size value is given. An error is generated if both `size` and `snapshotId` parameters are used.

Type: *SnapshotIdType*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in `CreateVolumeResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

volumeld

ID of the volume.

Type: *xs:string*

Examples

Example 1

The following example creates a volume of 2 GB.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVolume&Version=1&Timestamp=1318462897126&Expires=1318463197126&name=myVol2&size=1&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CreateVolumeResult" requestId="110"><volumeId>VOL-246b5c62-4072-41cf-885b-99d6c63583bd</volumeId></result>
```

Example 2

The following example creates a volume base on the specified snapshot ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVolume&Version=1&Timestamp=1320100009343&Expires=1320100309343&snapshotId=SNAP-7a717e39-fe67-4573-a93d-889b3446176b&name=myVol&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CreateVolumeResult" requestId="263"><volumeId>VOL-7d4993d6-f7f0-4611-b1a6-aca218746e64</volumeId></result>
```

See Also

- [DeleteVolumeRequest](#)
- [DescribeSnapshotsRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeAttributesRequest](#)
- [ImportVolumeRequest](#)
- [ModifyAttributesRequest](#)

DeleteAccessKeyRequest

Description

Action to unregister an access key from an account.

Request Parameters

accessKeyId

ID of the access key.

Type: *xs:string*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DeleteAccessKeyResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example unregisters the specified access key of a cloud user.

HTTP Request

```
https://<EnterpriseControllerHostname>/akm/?Action=DeleteAccessKey&Version=1&Times  
tamp=1318278941862&Expires=1318279241862&accessKeyId=AK_2
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result  
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="DeleteAccessKeyResult" requestId="134"/>
```

See Also

- [CreateAccessKeyAsObjectRequest](#)
- [CreateAccessKeyToFileRequest](#)
- [DescribeAccessKeysRequest](#)
- [DescribeAccountsRequest](#)
- [RegisterAccessKeyRequest](#)

DeleteDistributionGroupRequest

Description

Action to delete a distribution group from an account.

Request Parameters

distributionGroupId

ID of the distribution group.

Type: *DistributionGroupIdType*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DeleteDistributionGroupResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes a distribution group from an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteDistributionGroup&Version=1&Timestamp=1324407269441&Expires=1324407569441&distributionGroupId=DG-8f81381c-a559-4f5b-b45f-086e605a382b&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DeleteDistributionGroupResult" requestId="106"/>
```

See Also

- [CreateDistributionGroupRequest](#)
- [DescribeDistributionGroupsRequest](#)

DeleteKeyPairRequest

Description

Action to delete a key pair from an account.

Request Parameters

keyName

Name of the key pair.

Type: *xs:string*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DeleteKeyPairResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes a key pair from an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteKeyPair&Version=1&Timestamp=1318281994430&Expires=1318282294430&keyName=myKeyPair&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DeleteKeyPairResult" requestId="139"/>
```

See Also

- [CreateKeyPairAsObjectRequest](#)
- [CreateKeyPairToFileRequest](#)
- [DescribeKeyPairsRequest](#)
- [ImportKeyPairRequest](#)
- [RegisterKeyPairRequest](#)

DeleteSnapshotRequest

Description

Action to delete an existing snapshot from an account.

Request Parameters

snapshotId

ID of the snapshot.

Type: *SnapshotIdType*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DeleteSnapshotResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes the specified snapshot.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteSnapshot&Version=1&Times
tamp=1318624338550&Expires=1318624638550&snapshotId=SNAP-2f2039cd-943b-4072-9ded-e
96b54b7ca79&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DeleteSnapshotResult" requestId="234" />
```

See Also

- [CreateSnapshotRequest](#)
- [DescribeSnapshotsRequest](#)
- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

DeleteTagsRequest

Description

Action to delete one or more existing tags from a resource.

Request Parameters

resourceId

ID of the resource.

Type: *ResourceIdType*

Default: *None*

Required: *Yes*

tags

List of one or more tag names.

Type: *TagType*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DeleteTagsResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes the specified tag from a virtual network ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteTags&Version=1&Timestamp=1320102752960&Expires=1320103052960&resourceId=VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50&tags.1.name=myTag&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DeleteTagsResult" requestId="273"/>
```

See Also

- [CreateTagsRequest](#)

- [DescribeTagsRequest](#)

DeleteVnetRequest

Description

Action to delete a private virtual network from an account.

Request Parameters

vnet

ID of the virtual network.

Type: *VnetIdType*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DeleteVnetResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes the specified private virtual network.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteVnet&Version=1&Timestamp=1318283467620&Expires=1318283767620&vnet=VNET-00cd848c-771a-4091-b3f4-195a090bbc01&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DeleteVnetResult" requestId="142"/>
```

See Also

- [CreateVnetRequest](#)
- [DescribeVnetsRequest](#)
- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

DeleteVolumeRequest

Description

Action to delete a volume from an account.

Request Parameters

volumeId

ID of the volume.

Type: *VolumeIdType*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DeleteVolumeResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes the specified volume.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteVolume&Version=1&Timestamp=1318463692509&Expires=1318463992509&volumeId=VOL-3bbc8f1c-bb4c-4a70-a370-4b6c06d990dc&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DeleteVolumeResult" requestId="113"/>
```

See Also

- [CreateVolumeRequest](#)
- [DescribeAttributesRequest](#)
- [DescribeVolumesRequest](#)
- [ImportVolumeRequest](#)
- [ModifyAttributesRequest](#)

DeregisterServerTemplateRequest

Description

Action to delete a server template from an account.

Request Parameters

serverTemplateId

ID of the server template.

Type: *ServerTemplateIdType*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in *DeregisterServerTemplateResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes the specified server template.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeregisterServerTemplate&Version=1&Timestamp=1320098301701&Expires=1320098601701&serverTemplateId=TMPL-f089b985-f7fc-4b8a-a5f8-df8f44c95f3c&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DeregisterServerTemplateResult" requestId="256"/>
```

See Also

- [DescribeAttributesRequest](#)
- [DescribeServerTemplatesRequest](#)
- [DescribeTagsRequest](#)
- [RegisterServerTemplatesFromAssemblyRequest](#)
- [RegisterServerTemplateFromVserverRequest](#)
- [ModifyAttributesRequest](#)

DescribeAccessKeysRequest

Description

Action to display the available access keys for a cloud user.

Request Parameters

forUser

User name of a different cloud user. User using this option must have cloud administrator privileges.

Type: *GenericString*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in DescribeAccessKeysResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of access keys and their attributes.

Type: *DescribeAccessKeysResultItem*

Minimum: *0*

Maximum: *Unbounded*

forUser

User name of the other cloud user.

Type: *GenericString*

Examples

Example 1

The following example displays the available access keys for a cloud user.

HTTP Request

```
https://<EnterpriseControllerHostname>/akm/?Action=DescribeAccessKeys&Version=1&Timestamp=1318278142966&Expires=1318278442966
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeAccessKeysResult" requestId="133"><items><accessKeyId>AK_1</accessKeyId><account>ACC-f7fd508a-dad3-4866-a41a-bf8850163c3d</account></items></items><accessKeyId>AK_
```

```
2</accessKeyId><account>ACC-f7fd508a-dad3-4866-a41a-bf8850163c3d</account></items>  
<user>root</user></result>
```

See Also

- [CreateAccessKeyAsObjectRequest](#)
- [CreateAccessKeyToFileRequest](#)
- [DeleteAccessKeyRequest](#)
- [DescribeAccountsRequest](#)
- [RegisterAccessKeyRequest](#)

DescribeAccountsRequest

Description

Action to display account attributes for a cloud user. This action also allows filtering the accounts displayed by account ID.

Request Parameters

account

List of account IDs.

Type: *AccountIdType*

Default: *None*

Required: *No*

forUser

User name of a different cloud user. The user executing the action using this option must have cloud administrator privileges.

Type: *GenericString*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeAccountsResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the accounts and their attributes.

Type: *DescribeAccountsResultItem*

Minimum: *0*

Maximum: *Unbounded*

forUser

User name of the other cloud user.

Type: *GenericString*

Examples

Example 1

The following example displays the attributes of all existing accounts for a cloud user.

HTTP Request

```
https://<EnterpriseControllerHostname>/akm/?  
Action=DescribeAccounts&Version=1&Timestamp=1317054664930&Expires=1317054724930
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeAccountsResult"
requestId="102"><items><account>ACC-f7fd508adad3-
4866-a41a-bf8850163c3d</account><name>account0</name><description>Mockup
account for user clouduser0</description></items><items><account>ACC-
95adb073-d569-4a45-ac24-
f14eba689a86</account><name>account1</name><description>Mockup account for
user clouduser1</description></items><items><account>ACC-ea564db9-c0ee-43de-bb17-
55cfb9d105f2</account><name>account2</name><description>Mockup account for
user clouduser2</description></items><items><account>ACC-3e4a5732-1720-4348-b612-
43460b1c493d</account><name>account3</name><description>Mockup account for
user
clouduser3</description></items><items><account>ACC-2e2fdcb6-1568-435b-b40ce5418c2
0ab7b</
account><name>account4</name><description>Mockup account for
user clouduser4</description></items><user>root</user></result>
```

Example 2

The following example displays the attributes of the specified account.

HTTP Request

```
https://<EnterpriseControllerHostname>/akm/?
Action=DescribeAccounts&Version=1&Timestamp=1317055081608&Expires=1317055141608&ac
count=ACC-f7fd508a-dad3-4866-a41a-bf8850163c3d
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeAccountsResult"
requestId="107"><items><account>ACC-f7fd508adad3-
4866-a41a-bf8850163c3d</account><name>account0</name><description>Mockup
account for user clouduser0</description></items><user>root</user></result>
```

See Also

- [CreateAccessKeyAsObjectRequest](#)
- [CreateAccessKeyToFileRequest](#)
- [DeleteAccessKeyRequest](#)
- [DescribeAccessKeysRequest](#)
- [RegisterAccessKeyRequest](#)

DescribeAttributesRequest

Description

Action to display the value of one or more attributes of a resource.

This action is limited to the following resource types: server templates, snapshots, virtual networks, virtual servers, and volumes.

Request Parameters

resourceId

ID of a resource.

Type: *ResourceIdType*

Default: *None*

Required: *Yes*

attrNames

List of one or more attribute names of the resource.

Type: *GenericString*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in DescribeAttributesResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the resource attribute names and values.

Type: *ResourceAttribute*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example lists the values of the ID, size, status, and share attributes for the specified volume ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeAttributes&Version=1&Timestam
p=1318464041596&Expires=1318464341596&attrNames.4=shared&resourceId=VOL-246
b5c62-4072-41cf-885b-99d6c63583bd&attrNames.3=size&attrNames.2=status&attrNames.1=
id&AccessKeyId=AK_2&Signature=
SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
```

REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeAttributesResult"
requestId="115"><items><name>id</name><value>VOL-246b5c62-4072-41cf-885b-99d6c6358
3bd</value></items><items><name>status</name><value>OK</value></items><items><name
>size</name><value>0</value></items><items><name>shared</name></items></result>
```

See Also

- [DescribeServerTemplatesRequest](#)
- [DescribeSnapshotsRequest](#)
- [DescribeVnetsRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeVserversRequest](#)
- [ModifyAttributesRequest](#)

DescribeDistributionGroupsRequest

Description

Action to display the attributes of existing distribution groups for an account. This action allows filtering of distribution groups displayed by ID or by attribute names and values. Valid filter names are: id, name, description, status, vservers, and size.

Request Parameters

ids

List of distribution group IDs.

Type: *DistributionGroupIdType*

Default: *None*

Required: *No*

filters

List of distribution group attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeDistributionGroupsResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of distribution groups and their attributes.

Type: *DescribeDistributionGroupsResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example displays the available distribution groups for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeDistributionGroups&Version=1&Timestamp=1324407152926&Expires=1324407452926&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeDistributionGroupsResult"
requestId="105"><items><id>DG-8f81381c-a559-4f5b-b45f-086e605a382b</id><name>myDis
tributionGroup</name><status>OK</status><vservers>VSRV-062c548b-7346-4318-a802-a22
3288747e0</vservers><size>10</size></items><items><id>DG-9eb288c2-85e7-4392-80a6-d
1c1709de4cd</id><name>myDistributionGroup2</name><status>OK</status><size>10</size
></items></result>
```

See Also

- [CreateDistributionGroupRequest](#)
- [DeleteDistributionGroupRequest](#)

DescribeIpAddressesRequest

Description

Action to display allocated IP addresses attributes for an account. This action also allows filtering of the IP addresses displayed.

Request Parameters

filters

List of IP address attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeIpAddressesResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of allocated IP addresses and their attributes.

Type: *DescribeIpAddressesResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example lists all allocated IP addresses for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeIpAddresses&Version=1&
Timestamp=1320339663115&Expires=1320339963115&AccessKeyId=AK_
32&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeIpAddressesResult"
requestId="396"><items><ipAddress>10.6.0.3</ipAddress><vnet>VNET-6ea466f5-6e6b-415
9-adf3-8867473d4cf4</vnet></items><items><ipAddress>10.1.0.10</ipAddress><vnet>VNE
T-aaaaaaa8-bbb4-ccc4-ddd4-fffffffffff02</vnet></items><items><ipAddress>10.1.0.1</i
pAddress><vnet>VNET-aaaaaaa8-bbb4-ccc4-ddd4-fffffffffff02</vnet></items><items><ipA
ddress>10.1.0.0</ipAddress><vnet>VNET-aaaaaaa8-bbb4-ccc4-ddd4-fffffffffff02</vnet><
```

```
/items></result>
```

Example 2

The following example displays the allocated IP address attributes for the specified virtual network ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeIpAddresses&Version=1&Timestamp=1320339820989&Expires=1320340120989&filters.1.filterValue=VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4&filters.1.filterName=vnet&AccessKeyId=AK_32&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeIpAddressesResult" requestId="397"><items><ipAddress>10.6.0.3</ipAddress><vnet>VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4</vnet></items></result>
```

See Also

- [AllocateIpAddressesRequest](#)
- [DescribeVnetsRequest](#)
- [ReleaseIpAddressesRequest](#)

DescribeKeyPairsRequest

Description

Action to display attributes of existing key pairs for an account. This action allows filtering of key pairs displayed.

Request Parameters

keyNames

List of key pair names.

Type: *GenericString*

Default: *None*

Required: *No*

filters

List of key pair attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeKeyPairsResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of key pairs and their attributes.

Type: *DescribeKeyPairsResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example displays the available key pairs for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeKeyPairs&Version=1&Timestamp=1320361610130&Expires=1320361910130&AccessKeyId=AK_32&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
```

```
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="DescribeKeyPairsResult"  
requestId="437"><items><name>myKeyPairK</name><fingerprint>02:ed:56:e2:e4:a7:c9:ed  
:3f:e3:91:34:fc:c7:1a:3a</fingerprint></items><items><name>myKeyPair2</name><finge  
rprint>c1:0d:00:86:e9:99:1b:e3:43:62:71:0f:00:17:d4:50</fingerprint></items><items  
><name>myKeyPair3</name><fingerprint>b6:a3:41:2e:fa:f8:24:c2:f6:db:85:6f:e5:1f:8b:  
fb</fingerprint></items></result>
```

See Also

- [CreateKeyPairAsObjectRequest](#)
- [CreateKeyPairToFileRequest](#)
- [DeleteKeyPairRequest](#)
- [ImportKeyPairRequest](#)
- [RegisterKeyPairRequest](#)

DescribeServerTemplatesRequest

Description

Action to display attributes of available server templates for an account. This action also allows filtering of server templates displayed.

Request Parameters

ids

List of one or more server template IDs.

Type: *ServerTemplateIdType*

Default: *None*

Required: *No*

filters

List of server template attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeServerTemplatesResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of server templates and their attributes.

Type: *DescribeServerTemplatesResultItem*

Examples

Example 1

The following example lists all the available server templates for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeServerTemplates&Version=1&Timestamp=1320088801876&Expires=1320089101876&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeServerTemplatesResult"
```

```
requestId="245"><items><id>TMPL-e7df3178-12ea-48cc-8a1a-6b3f8256dd0f</id><name>SendReceiveImage.tar.gz</name><status>OK</status><size>-1</size><public>>false</public><imageType>VMTemplate</imageType><readOnly>>false</readOnly></items><items><id>TMPL-48cfe488-b93d-4085-aa2e-0a3b39d511cd</id><name>OVM_EL52_jeos_i386_PVM_WebLogic10gR3_v10.tar.gz</name><description>Import URLs: [http://example.oracle.com/vm-templates/OVM_EL52_jeos_i386_PVM_WebLogic10gR3_v10.tar.gz]</description><status>OK</status><size>1</size><public>>false</public><imageType>VMTemplate</imageType><readOnly>>false</readOnly></items></result>
```

Example 2

The following example displays the attributes of the specified server template ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeServerTemplates&Version=1&Timestamp=1320088942728&Expires=1320089242728&ids.1=TMPL-e7df3178-12ea-48cc-8a1a-6b3f8256dd0f&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeServerTemplatesResult" requestId="246"><items><id>TMPL-e7df3178-12ea-48cc-8a1a-6b3f8256dd0f</id><name>SendReceiveImage.tar.gz</name><status>OK</status><size>1</size><public>>false</public><imageType>VMTemplate</imageType><readOnly>>false</readOnly></items></result>
```

See Also

- [DeregisterServerTemplateRequest](#)
- [RegisterServerTemplatesFromAssemblyRequest](#)
- [RegisterServerTemplateFromUrlRequest](#)
- [RegisterServerTemplateFromVserverRequest](#)
- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

DescribeSnapshotsRequest

Description

Action to display the attributes of the available snapshots for an account. This action also allows filtering of the snapshots displayed.

Request Parameters

ids

List of one or more snapshot IDs.

Type: *GenericString*

Default: *None*

Required: *No*

filters

List of snapshot attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeSnapshotsResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the snapshots and their attributes.

Type: *DescribeSnapshotsResultItem*

Examples

Example 1

The following example lists the attributes of all the available snapshots for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeSnapshots&Version=1&Timestamp=1318467903784&Expires=1318468203784&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeSnapshotsResult"
```

```
requestId="121"><items><id>SNAP-2f2039cd-943b-4072-9ded-e96b54b7ca79</id><name>myS  
napshot2</name><status>OK</status></items><items><id>SNAP-d743e90c-53c5-4b01-a297-  
e2c944755c48</id><name>mySnapshot</name><status>OK</status></items></result>
```

Example 2

The following example displays the attributes for the specified snapshot ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeSnapshots&Version=1&Ti  
mestamp=1318468567145&Expires=1318468867145&ids.1=SNAP-2f2039cd-943b-4072-9ded-e96  
b54b7ca79&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_  
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result  
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="DescribeSnapshotsResult"  
requestId="122"><items><id>SNAP-2f2039cd-943b-4072-9ded-e96b54b7ca79</id><name>myS  
napshot2</name><status>OK</status></items></result>
```

See Also

- [AttachVolumesToVserverRequest](#)
- [CreateVolumeRequest](#)
- [DeleteVolumeRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeAttributesRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [ModifyAttributesRequest](#)

DescribeTagsRequest

Description

Action to display the tag attributes of the resources with an associated tag. This action also allows filtering of the tags or resources displayed.

Request Parameters

ids

List of one or more resource IDs.

Type: *ResourceIdType*

Default: *None*

Required: *No*

filters

List of resource attributes or tag names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeTagsResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the resource tags and their attributes.

Type: *DescribeTagsResultItem*

Examples

Example 1

The following example lists all tags available for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeTags&Version=1&Timestamp=1320101837081&Expires=1320102137081&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeTagsResult"
```

```
requestId="270"><items><resourceId>VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50</resourceId><name>oc.internal.oracle.cloud.security.group.hidden</name><value>private</value></items><items><resourceId>VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50</resourceId><name>myTag</name><value>myTagValue</value></items><items><resourceId>VNET-2b8b6c91-065c-4645-9cdb-edb3afd92524</resourceId><name>oc.internal.oracle.cloud.security.group.hidden</name><value>private</value></items><items><resourceId>VNET-07e74ad3-1ab1-4188-915d-cbf6242a1eeb</resourceId><name>oc.internal.oracle.cloud.security.group.hidden</name><value>private</value></items></result>
```

Example 2

The following example displays existing tag attributes for the specified virtual network ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeTags&Version=1&Timestamp=1320102284357&Expires=1320102584357&ids.1=VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeTagsResult" requestId="272"><items><resourceId>VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50</resourceId><name>oc.internal.oracle.cloud.security.group.hidden</name><value>private</value></items><items><resourceId>VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50</resourceId><name>myTag</name><value>myTagValue</value></items></result>
```

See Also

- [CreateTagsRequest](#)
- [DeleteTagsRequest](#)

DescribeVdcCapabilitiesRequest

Description

Action to display the vDC capabilities for an account. This action also allows filtering of the vDC capabilities displayed.

Request Parameters

filters

List of the virtual data center capability names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeVdcCapabilitiesResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the vDC capabilities and their values.

Type: *DescribeVdcCapabilitiesResultItem*

Minimum: *1*

Maximum: *Unbounded*

Examples

Example 1

The following example lists the vDC capabilities for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVdcCapabilities&Version=1&Timestamp=1320085185647&Expires=1320085485647&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="DescribeVdcCapabilitiesResult" requestId="243"><items><name>VirtualizationType</name><value>OVM</value></items><items><name>VirtualizationVersion</name><value>3.0.2</value></items><items><name>ProcessorArch</name><value>Default_Intel_F6_M23</value></items><items><name>ProcessorVersion</name><value>Default_Intel_
```

```
Family:6_
Model:23</value></items><items><name>DistributionGroupSupport</name><value>disabled</value></items><items><name>HighAvailabilityUserControl</name><value>disabled</value></items><items><name>HighAvailabilityDefault</name><value>true</value></items>
</result>
```

Example 2

The following example lists the virtualization type of the vDC for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVdcCapabilities&Version=1&Timestamp=1320085633535&Expires=1320085933535&filters.1.filterName=VirtualizationType&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVdcCapabilitiesResult"
requestId="244"><items><name>VirtualizationType</name><value>OVM</value></items></result>
```

DescribeVnetsRequest

Description

Action to display the attributes of the available virtual networks for an account. This action also allows filtering of the virtual networks displayed.

Request Parameters

ids

List of one or more virtual network IDs.

Type: *VnetIdType*

Default: *None*

Required: *No*

filters

List of virtual network attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeVnetsResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the virtual networks and their attributes.

Type: *DescribeVnetsResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example lists the available virtual networks for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?  
Action=DescribeVnets&Version=1&Timestamp=1317048258687&Expires=1317048318687&Access  
s  
KeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_READABILITY_OF_THE  
_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:type="DescribeVnetsResult" requestId="256"><items><id>VNET-cbd75904-9d34-4ec8-
86b9-d917712abaaa</id><name>stevennet</name><description>stevennet</description><s
tatus>OK</status><tags><name>oc.internal.oracle.cloud.security.group.hidden</name>
<value>private</value></tags><ipAddress>192.168.2.0/24</ipAddress></items><items><
id>VNET-c41e4523-0578-4b48-a9f6-77d766b053d9</id><name>vnet46</name><description>v
net46</description><status>OK</status><tags><name>oc.internal.oracle.cloud.securit
y.group.hidden</name><value>private</value></tags><ipAddress>192.168.4.0/24</ipAdd
ress></items><items><id>VNET-77d2b71b-a5ff-4d76-bdc4-fbd9317bbdb8</id><name>vnet3<
/name><description>vnet3</description><status>OK</status><tags><name>oc.internal.o
racle.cloud.security.group.hidden</name><value>private</value></tags><ipAddress>19
2.168.1.0/24</ipAddress></items><items><id>VNET-5d74972a-bcdd-4714-8c7fb67d8010f13
c</id><name>vnet123</name><description>vnet1</description><status>OK</status><ta
gs><name>oc.internal.oracle.cloud.security.group.hidden</name><value>private</valu
e></tags><ipAddress>192.168.0.0/24</ipAddress></items><items><id>VNET-7b91f019-b50
d-4051-8028-7ed2b0f5d767</id><name>vnet45</name><description>vnet45</description><
status>OK</status><tags><name>oc.internal.oracle.cloud.security.group.hidden</name
><value>private</value></tags><ipAddress>192.168.3.0/24</ipAddress></items><items>
<id>VNET-9058613f-efaa-42f4-bc96-9583ec39a481</id><name>vnet47</name><description>
vnet47</description><status>OK</status><tags><name>oc.internal.oracle.cloud.securi
ty.group.hidden</name><value>private</value></tags><ipAddress>192.168.5.0/24</ipAd
dress></items></result>
```

Example 2

The following example displays the attributes for the specified virtual network.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?
Action=DescribeVnets&Version=1&Timestamp=1317049897108&Expires=1317049957108&ids.1
=VNET-cbd75904-9d34-4ec8-86b9-d917712abaaa&AccessKeyId=AK_2&Signature=SIGNATURE_
HAS_BEEN_REMOVED_FOR_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVnetsResult" requestId="262"><items><id>VNET-cbd75904-9d34-4ec8-
86b9-d917712abaaa</id><name>stevennet</name><description>stevennet</description><s
tatus>OK</status><tags><name>oc.internal.oracle.cloud.security.group.hidden</name>
<value>private</value></tags><ipAddress>192.168.2.0/24</ipAddress></items></result
>
```

See Also

- [CreateVnetRequest](#)
- [DeleteVnetRequest](#)
- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

DescribeVolumesRequest

Description

Action to display the attributes of the available volumes for an account. This action also allows filtering of volumes displayed. Valid filter names are: description, name, shared, status, id, and vservers.

Request Parameters

ids

List of one or more volume IDs.

Type: *VolumeIdType*

Default: *None*

Required: *No*

filters

List of volume attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in DescribeVolumesResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of volumes and their attributes.

Type: *DescribeVolumesResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example displays the attributes of the available volumes for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVolumes&Version=1&Time
stamp=1318463287921&Expires=1318463587921&AccessKeyId=AK_2&Signature=SIGNATURE_
HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVolumesResult"
requestId="111"><items><id>VOL-246b5c62-4072-41cf-885b-99d6c63583bd</id><name>myVo
l2</name><status>OK</status><size>0</size><shared>>false</shared><vservers>VSRV-76e
d9c11-d5d5-4418-833d-b2025c7bfdeb</vservers></items><items><id>VOL-3bbc8f1c-bb4c-4
a70-a370-4b6c06d990dc</id><name>myVol</name><status>OK</status><size>0</size><shar
ed>>false</shared><vservers>VSRV-5100c512-fc39-4a59-ade9-653ale041c10</vservers></i
tems></result>
```

Example 2

The following example displays the attributes of the specified volume ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVolumes&Version=1&Time
stamp=1318463504971&Expires=1318463804971&ids.1=VOL-3bbc8f1c-bb4c-4a70-a370-4b6c06
d990dc&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_
OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVolumesResult"
requestId="112"><items><id>VOL-3bbc8f1c-bb4c-4a70-a370-4b6c06d990dc</id><name>myVo
l</name><status>OK</status><size>0</size><shared>>false</shared><vservers>VSRV-5100
c512-fc39-4a59-ade9-653ale041c10</vservers></items></result>
```

See Also

- [AttachVolumesToVserverRequest](#)
- [CreateVolumeRequest](#)
- [DeleteVolumeRequest](#)
- [DescribeAttributesRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [ImportVolumeRequest](#)
- [ModifyAttributesRequest](#)

DescribeVserverMetricsRequest

Description

Action to display vServer metrics.

This action also allows filtering of the vServers displayed. Valid filter names are: description, name, status, id, and cpuUsage.

Request Parameters

ids

List of one or more vServer IDs.

Type: *VserverIdType*

Default: *None*

Required: *No*

filters

List of vServer attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in DescribeVserversResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the vServers and their attributes.

Type: *DescribeVserverMetricsResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example displays metrics of all vServers available for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVserverMetrics&Version=1&Timestamp=1320105338731&Expires=1320105638731&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVserversResult"
requestId="276"><items><id>VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64</id><name>tes
tvserver3</name><description>Oracle VM Virtual
Machine</description><status>RUNNING</status><cpuUsage>0.1508</cpuUsage></items><i
tems><id>VSRV-71f63b37-d61e-4900-a1a5-2b7d51b029c9</id><name>testvserver5</name><d
escription>Oracle VM Virtual
Machine</description><status>RUNNING</status><cpuUsage>0.067</cpuUsage></items></r
esult>
```

Example 2

The following example displays the vServer metrics for the specified vServer ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVservers&Version=1&Tim
estamp=1320105445075&Expires=1320105745075&ids.1=VSRV-2e4f6688-2c57-4f41-8d3d-83d2
d71d1a64&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVserversResult"
requestId="277"><items><id>VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64</id><name>tes
tvserver3</name><description>Oracle VM Virtual
Machine</description><status>RUNNING</status><cpuUsage>0.1508</cpuUsage></items></
result>
```

See Also

- [AttachVolumesToVserverRequest](#)
- [DescribeAttributesRequest](#)
- [DescribeVserversRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [ModifyAttributesRequest](#)
- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)
- [SendMessageToVserverRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

DescribeVserversRequest

Description

Action to display vServer attributes for an account.

This action also allows filtering of vServers displayed. Valid filter names are: *description*, *name*, *status*, *id*, *serverTemplateId*, *keyName*, *vserverType*, *distributionGroup*, *volumes*, *vcpu*, *memoryMb*, *dedicatedStorageMb*, and *attachedStorageMb*.

Request Parameters

ids

List of one or more vServer IDs.

Type: *VserverIdType*

Default: *None*

Required: *No*

filters

List of vServer attribute names and values.

Type: *FilterItem*

Default: *None*

Required: *No*

Result Elements

The result elements are contained in *DescribeVserversResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of the vServers and their attributes.

Type: *DescribeVserversResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1

The following example displays attributes of all vServers available for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVservers&Version=1&Timestamp=1320105338731&Expires=1320105638731&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVserversResult"
requestId="276"><items><id>VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64</id><name>tes
tvserver3</name><description>Oracle VM Virtual
Machine</description><status>RUNNING</status><vnets>VNET-025a35bf-828d-4e3f-a0f4-7
6f3d9a73c42</vnets><ipAddresses>192.168.0.18</ipAddresses><serverTemplateId>TMPL-5
57952b6-0b00-4a00-a2ca-ada480d99cc6598</serverTemplateId><ha>true</ha><volumes>VOL
-dc80ef69-109d-4e7d-968d-2fa90b68db1b</volumes><vcpu>2</vcpu><memoryMb>8192</memor
yMb><dedicatedStorageMb>6443499520</dedicatedStorageMb><attachedStorageMb>32212254
72</attachedStorageMb></items><items><id>VSRV-71f63b37-d61e-4900-a1a5-2b7d51b029c9
</id><name>testvserver5</name><description>Oracle VM Virtual
Machine</description><status>RUNNING</status><vnets>VNET-025a35bf-828d-4e3f-a0f4-7
6f3d9a73c42</vnets><ipAddresses>192.168.0.19</ipAddresses><serverTemplateId>TMPL-5
57952b6-0b00-4a00-a2ca-ada480d99cc6597</serverTemplateId><ha>true</ha><vcpu>1</vcp
u><memoryMb>4096</memoryMb><dedicatedStorageMb>6443499520</dedicatedStorageMb><att
achedStorageMb>0</attachedStorageMb></items></result>
```

Example 2

The following example displays the vServer attributes for the specified vServer ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVservers&Version=1&Tim
estamp=1320105445075&Expires=1320105745075&ids.1=VSRV-2e4f6688-2c57-4f41-8d3d-83d2
d71d1a64&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVserversResult"
requestId="277">items><id>VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64</id><name>test
vserver3</name><description>Oracle VM Virtual
Machine</description><status>RUNNING</status><vnets>VNET-025a35bf-828d-4e3f-a0f4-7
6f3d9a73c42</vnets><ipAddresses>192.168.0.18</ipAddresses><serverTemplateId>TMPL-5
57952b6-0b00-4a00-a2ca-ada480d99cc6598</serverTemplateId><ha>true</ha><volumes>VOL
-dc80ef69-109d-4e7d-968d-2fa90b68db1b</volumes><vcpu>2</vcpu><memoryMb>8192</memor
yMb><dedicatedStorageMb>6443499520</dedicatedStorageMb><attachedStorageMb>32212254
72</attachedStorageMb></items></result>
```

See Also

- [AttachVolumesToVserverRequest](#)
- [DescribeAttributesRequest](#)
- [DescribeVserverMetricsRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [ModifyAttributesRequest](#)
- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)

- [RunVserverRequest](#)
- [RunVserversRequest](#)
- [SendMessageToVserverRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

DescribeVserverTypesRequest

Description

Action to display the permitted vServer types for an account.

Request Parameters

There are no request parameters for this action.

Result Elements

The result elements are contained in DescribeVserverTypesResult..

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

items

List of vServer types and their attributes.

Type: *DescribeVserverTypesResultItem*

Minimum: *1*

Examples

Example 1

The following example lists all available vServer types for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVserverTypes&Version=1
&Timestamp=1318625453358&Expires=1318625753358&AccessKeyId=AK_
2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="DescribeVserverTypesResult"requestId="236"><items><id>small</id><name>sm
all</name><description>small instance
type</description><memorySize>1073741824</memorySize><storageSize>10737418240</sto
rageSize><vcpu>1</vcpu></items><items><id>medium</id><name>medium</name><descripti
on>medium instance type</description><memorySize>4294967296</memorySize>
<storageSize>107374182400</storageSize><vcpu>2</vcpu></items><items><id>large</id>
<name>large</name><description>large instance type</description>
<memorySize>17179869184</memorySize><storageSize>1073741824000</storageSize><vcpu>
4</vcpu></items></result>
```

See Also

- [DescribeVserversRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)

DetachVolumesFromVserverRequest

Description

Action to detach one or more volumes from a vServer.

Request Parameters

vserverId

ID of the vServer.

Type: *VserverIdType*

Default: *None*

Required: *Yes*

volumeIds

List of one or more volume IDs.

Type: *VolumeIdType*

Default: *None*

Required: *Yes*

force

Option to force detach.

Type: *xs:boolean*

Default: *0 (false)*

Required: *No*

Result Elements

The result elements are contained in `DetachVolumesFromVserverResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example detaches the specified volumes from the vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DetachVolumesFromVserver&Version=1&Timestamp=1321385229679&Expires=1321385529679&volumeIds.1=VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15&force=true&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
```

```
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="DetachVolumesFromVserverResult" requestId="156"/>
```

See Also

- [AttachVolumesToVserverRequest](#)
- [CreateVolumeRequest](#)
- [DeleteVolumeRequest](#)
- [DescribeAttributesRequest](#)
- [DescribeVolumesRequest](#)
- [ImportVolumeRequest](#)
- [ModifyAttributesRequest](#)

ImportKeyPairRequest

Description

Action to import an existing key pair to an account.

When importing a key pair, the cloud user supplies the file that stores the public key and then the public key is registered. The cloud user manages the private key himself.

ImportKeyPair is a client-side action, it internally makes a call to the RegisterKeyPair action.

Request Parameters

keyName

Name of the key pair.

Type: *xs:string*

Default: *None*

Required: *Yes*

keyFileName

Name of the file that stores the public key of the key pair to import.

Type: *GenericString*

Default: *None*

Required: *Yes*

Result Elements

The result elements are contained in ImportKeyPairResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

keyName

Name of the key pair.

Type: *xs:string*

keyFingerprint

Key pair fingerprint.

Type: *xs:string*

Examples

Example 1

The following example imports and registers a key pair using the public key in the specified key file.

HTTP Request

`https://<EnterpriseControllerHostname>/iaas/?Action=RegisterKeyPair&Version=1&Time`

```
stamp=1324421510757&Expires=1324421810757&publicKey=ssh-rsa+AAAAB3NzaC1yc2EAAAABIwAAQEAAtJnFD8INGLtM%2FQIlxkJh4t2R4%2FtTmPUDzMIRP%0A&keyName=myKeyPair&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RegisterKeyPairResult" requestId="110"><keyName>myKeyPair</keyName><keyFingerprint>f4:20:bf:ad:ed:e7:0a:1a:a5:74:f9:80:72:bb:74:b4</keyFingerprint></result>
```

See Also

- [CreateKeyPairAsObjectRequest](#)
- [CreateKeyPairToFileRequest](#)
- [DeleteKeyPairRequest](#)
- [DescribeKeyPairsRequest](#)
- [RegisterKeyPairRequest](#)

ImportVolumeRequest

Description

Action to import a volume to an account.

Request Parameters

name

Name of the volume.

Type: *GenericString*

Required: *Yes*

description

Description of the volume.

Type: *GenericString*

Required: *No*

url

URL of the volume to import.

Type: *URLType*

Required: *No*

shared

Flag to indicate if the volume is shared.

Type: *xs:boolean*

Default: *0 (false)*

Required: *No*

Result Elements

The result elements are contained in `ImportVolumeResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

volumeId

ID of the volume.

Type: *xs:string*

Examples

Example 1

The following example imports a shared volume for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=ImportVolume&Version=1&Timestamp=1324421943593&Expires=1324422243593&shared=true&name=myVolumeImported&url=http%3A%2F%2Foracle.com%2FES%2FOVM%2Fvolume%2Fvolume1.img&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ImportVolumeResult" requestId="111"><volumeId>VOL-b71cb98a-52db-49cc-aba4-66a1dfb4d13f</volumeId></result>
```

See Also

- [AttachVolumesToVserverRequest](#)
- [CreateVolumeRequest](#)
- [DeleteVolumeRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeAttributesRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [ModifyAttributesRequest](#)

ModifyAttributesRequest

Description

Action to modify one or more attribute values for a resource.

Request Parameters

resourceId

ID of a resource.

Type: *ResourceIdType*

Default: *None*

Required: *Yes*

attributes

List of one or more attribute names and values of a resource.

Type: *ResourceAttribute*

Required: *Yes*

Result Elements

The result elements are contained in `ModifyAttributesResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example modifies the name and description for the specified volume ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=ModifyAttributes&Version=1&Timestamp=1318464546890&Expires=1318464846890&resourceId=VOL-246b5c62-4072-41cf-885b-99d6c63583bd&attributes.2.value=myNewDescription&attributes.2.name=description&attributes.1.name=name&attributes.1.value=myNewName&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ModifyAttributesResult" requestId="116"/>
```

See Also

- [DescribeAttributesRequest](#)
- [DescribeServerTemplatesRequest](#)

- [DescribeSnapshotsRequest](#)
- [DescribeVnetsRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeVserversRequest](#)

RebootVserversRequest

Description

Action to reboot one or more existing vServers.

Request Parameters

vserverIds

One or more vServer IDs.

Type: *VserverIdType*

Required: *Yes*

Result Elements

The result elements are contained in `RebootVserversResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example reboots a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=RebootVservers&Version=1&Times  
tamp=1321380470326&Expires=1321380770326&vserverIds.1=VSRV-0fb57293-347c-4717-96ef  
-6dd23154596f&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_  
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result  
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="RebootVserversResult" requestId="121"/>
```

See Also

- [ReceiveMessageFromVserverRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)
- [SendMessagesToVserverRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

ReceiveMessageFromVserverRequest

Description

Action to read a message from a vServer.

This action reads a message by name from the common buffer in the virtualization layer. This action may return an error if the vServer is not in an appropriate state to receive messages.

Request Parameters

vserverId

ID of the vServer.

Type: *VserverIdType*

Required: *Yes*

key

Name of the key for the message to retrieve.

Type: *GenericString*

Required: *Yes*

Result Elements

The result elements are contained in `ReceiveMessageFromVserverResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

value

Value of the message.

Type: *xs:string*

Examples

Example 1

The following example receives a message from a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=ReceiveMessageFromVserver&Version=1&Timestamp=1321381585564&Expires=1321381885564&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&key=myMessageKey&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ReceiveMessageFromVserverResult"
```

```
requestId="123"<value>myMessageFromVserver</value></result>
```

See Also

- [RebootVserversRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)
- [SendMessageToVserverRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

RegisterAccessKeyRequest

Description

Action to register an access key of a cloud user to the specified account. This action also allows the creation of an access key on behalf of another cloud user. In that case, the user running this action requires cloud administrator privileges.

Request Parameters

publicKey

Public key of the access key.

Type: *xs:string*

Required: *Yes*

account

ID of the account.

Type: *AccountIdType*

Required: *Yes*

forUser

Name of another cloud user.

Type: *GenericString*

Required: *No*

Result Elements

The result elements are contained in RegisterAccessKeyResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

accessKeyId

ID of the access key.

Type: *xs:string*

Examples

Example 1

The following example registers an access key for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/akm/?
Action=RegisterAccessKey&Version=1&Timestamp=1317055640451&Expires=1317055700451&p
ub
licKey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAE3XtkY0aBtOSFZcmk
%2FPjjiMj7xyGCzRB2I%0AwQXV
%2BQiY5La1Ppj8fSxrs1t85Hy2%2FUY8gfVYy3peGmlmko0xtFfP90ACxKAH7Z8%2B8LggPT6w
DxY%0Ax1kOF9k80M9fHxXrfWNfxfw87yd%2FNZdeZOvgRxw8B
```

```
%2BsuHmAV5HyRRFKuiQFko7EYVAijFIhRv7ez%0AThKNmP  
%2BSrOGvPuUoG035TDloSbdpQ08ZGurvIdGevcJpM3IfOEy5tqyfU%2FdvtalH%2FyhabUjCZN6E  
%0AqGs2t9C75D9sgVy%2FykbDiXYOorRV3wUoatB43YDPpx5TjpJZIogeF1vCVETDGg6sKsv1  
%2FpqRU3FH%0AWTHCOwIDAQAB&account=ACC-f7fd508a-dad3-4866-a41a-bf8850163c3d
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result  
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="RegisterAccessKeyResult"  
requestId="110"><accessKeyId>AK_1</accessKeyId></result>
```

See Also

- [CreateAccessKeyAsObjectRequest](#)
- [CreateAccessKeyToFileRequest](#)
- [DeleteAccessKeyRequest](#)
- [DescribeAccessKeysRequest](#)
- [DescribeAccountsRequest](#)

RegisterKeyPairRequest

Description

Action to register a key pair for an account.

Request Parameters

keyName

Name of the key pair.

Type: *xs:string*

Required: *Yes*

publicKey

Public key of the key pair.

Type: *xs:string*

Required: *Yes*

Result Elements

The result elements are contained in RegisterKeyPairResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

keyFingerprint

Key pair fingerprint.

Type: *xs:string*

Examples

Example 1

The following example registers a key pair for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterKeyPair&Version=1&Time
stamp=1318279380444&Expires=1318279680444&publicKey=ssh-rsa+AAAAB3NzaC1yc2EAAAADAQ
ABAAABAQDg2eienGE4vEMSMcVMbYbC8z2q%2Fvzh3H6AanlJ6B4udseK%0A8CpaHJ23eGwcjcgAmuZCJ%2
FOoHUA2dn2PNPuK6g%2BZndR8wVaaQT89eWDZx9oaf0%2F2Eg%2FLeKJ3moVH%0AvIYvFB9aFCpa4H%2BO
mLfM%2FmfQ4CYedfo0r0jxCCB0YLo0876LQqK5X%2BtgRXwbAbPH2Mzbzp%2FzzdkQ%0ArsBqSgUQ%2B1V
4LkN6TQe06P5a2QYI1UhRXwUorTnbXczGq9zEJJ7ef%2F74xIQZfAipkYkyGgktsXrM%0A%2F%2Bs789v9
ipaDB5B26y3aqjIdvW4ZLDvuGXPs60aiUfj2WGIqx0KSVL%2FyB%2FtK1WbuZYw1+IaaS-Generated&ke
yName=myKeyPairK&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="RegisterKeyPairResult"
```

```
requestId="137"><keyFingerprint>65:3d:ac:81:90:21:2c:4e:65:78:99:b2:37:13:00:93</keyFingerprint></result>
```

See Also

- [CreateKeyPairAsObjectRequest](#)
- [CreateKeyPairToFileRequest](#)
- [DeleteKeyPairRequest](#)
- [DescribeKeyPairsRequest](#)
- [ImportKeyPairRequest](#)

RegisterServerTemplateFromUrlRequest

Description

Action to register a server template from a URL.

Request Parameters

name

Name of the server template.

Type: *GenericString*

Required: *Yes*

description

Description of the server template.

Type: *GenericString*

Required: *No*

url

URL of the server template.

Type: *URLType*

Required: *Yes*

Result Elements

The result elements are contained in RegisterServerTemplateFromUrlResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

serverTemplateId

ID of the server template.

Type: *xs:string*

Examples

Example 1

The following example registers a server template for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplateFromUrl&Version=1&Timestamp=1320096741216&Expires=1320097041216&name=myST&url=http%3A%2F%2Fexample.oracle.com%2Fvm-templates%2FOVM_EL52_jeos_i386_PVM_WebLogic10gR3_v10.tar.gz&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
```

```
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="RegisterServerTemplateFromUrlResult"  
requestId="248"><serverTemplateId>TMPL-f089b985-f7fc-4b8a-a5f8-df8f44c95f3c</serverTemplateId></result>
```

See Also

- [DeregisterServerTemplateRequest](#)
- [DescribeAttributesRequest](#)
- [DescribeServerTemplatesRequest](#)
- [ModifyAttributesRequest](#)
- [RegisterServerTemplateFromVserverRequest](#)
- [RegisterServerTemplatesFromAssemblyRequest](#)

RegisterServerTemplateFromVserverRequest

Description

Action to register a server template from a vServer.

Request Parameters

name

Name of the server template.

Type: *GenericString*

Required: *Yes*

description

Description of the server template.

Type: *GenericString*

Required: *No*

vserverId

ID of the vServer.

Type: *VserverIdType*

Required: *Yes*

Result Elements

The result elements are contained in RegisterServerTemplateFromVserverResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

serverTemplateId

ID of the server template.

Type: *xs:string*

Examples

Example 1

The following example creates a server template for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplateFromVserver&Version=1&Timestamp=1324422880552&Expires=1324423180552&name=mySTfromVserver&vserverId=VSRV-fdba0b48-6e1c-4f41-bde4-2c739dfeeb2f&AccessKeyId=AK_2&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
```

```
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="RegisterServerTemplateFromVserverResult"  
requestId="258"><serverTemplateId>TMPL-f047b985-f7fc-4b8a-a5f8-df8f44c95f5a</serverTemplateId></result>
```

See Also

- [DeregisterServerTemplateRequest](#)
- [DescribeAttributesRequest](#)
- [DescribeServerTemplatesRequest](#)
- [ModifyAttributesRequest](#)
- [RegisterServerTemplateFromUrlRequest](#)
- [RegisterServerTemplatesFromAssemblyRequest](#)

RegisterServerTemplatesFromAssemblyRequest

Description

Action to register a set of server templates from an assembly.

Request Parameters

name

Name of the assembly.

Type: *GenericString*

Required: *Yes*

description

Description of the assembly.

Type: *GenericString*

Required: *No*

url

URL of the assembly.

Type: *URLType*

Required: *Yes*

Result Elements

The result elements are contained in RegisterServerTemplatesFromAssemblyResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

serverTemplateIds

IDs of the server templates.

Type: *xs:string*

Examples

Example 1

The following example creates a set of server templates for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplatesFromAssembly&Version=1&Timestamp=1320097901377&Expires=1320098201377&name=myStk2&url=http%3A%2F%2Fadc4120293.us.oracle.com%3A8888%2FmyAssembly.ova&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
```

```
xmlns="http://www.oracle.com/xml/ns/iaas"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:type="RegisterServerTemplatesFromAssemblyResult"  
requestId="254"><serverTemplateIds>ASSM-fe68b60a-b28b-45fa-8115-0801e3df67d5</serv  
erTemplateIds></result>
```

See Also

- [DeregisterServerTemplateRequest](#)
- [DescribeAttributesRequest](#)
- [DescribeServerTemplatesRequest](#)
- [ModifyAttributesRequest](#)
- [RegisterServerTemplateFromUrlRequest](#)
- [RegisterServerTemplateFromVserverRequest](#)

ReleaseIpAddressesRequest

Description

Action to release one or more allocated IP addresses from a virtual network.

Request Parameters

vnet

ID of the virtual network.

Type: *VnetIdType*

Required: *Yes*

ipAddresses

One or more allocated IP addresses.

Type: *GenericString*

Required: *Yes*

Result Elements

The result elements are contained in `ReleaseIpAddressesResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example releases the specified IP addresses from the virtual network.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=ReleaseIpAddresses&Version=1&Timestam
p=1320341801846&Expires=1320342101846&ipAddresses.2=10.6.0.10&ipAddresses.1
=10.6.0.13&vnet=VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4&AccessKeyId=AK_
32&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ReleaseIpAddressesResult" requestId="402"/>
```

See Also

- [AllocateIpAddressesRequest](#)
- [CreateVnetRequest](#)
- [DescribeIpAddressesRequest](#)
- [DescribeVnetsRequest](#)

RunVserverRequest

Description

Action to create and start a vServer.

Request Parameters

name

Name of the vServer.

Type: *GenericString*

Required: *Yes*

description

Description of the vServer.

Type: *GenericString*

Required: *No*

serverTemplateId

ID of a server template.

Type: *ServerTemplateIdType*

Required: *Yes*

ipAddresses

List of one or more reserved IP addresses.

Type: *GenericString*

Required: *Yes*

keyName

Name of a key pair.

Type: *GenericString*

Required: *No*

vserverType

ID of the vServer type.

Type: *GenericString*

Required: *Yes*

vnets

List of one or more virtual network IDs.

Type: *VnetIdType*

Required: *Yes*

volumes

List of one or more volume IDs.

Type: *VolumeIdType*

Required: *No*

distGroup

ID of the distribution group.

Type: *DistributionGroupIdType*

Required: *No*

ha

Flag to indicate if high availability is enabled for the vServer. The flag is boolean; possible values are true or false.

Type: *GenericString*

Required: *No*

messages

List of messages sent as key-value pairs to the vServer.

Type: *Message*

Required: *No*

hostname

A host name to set or override the internal host name for the vServer. It must be RFC 1123 compliant.

Type: *HostnameType*

Required: *No*

rootPassword

A password to set or override the root password for the vServer.

Type: *GenericString*

Required: *No*

Result Elements

The result elements are contained in *RunVserverResult*.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

vserverId

ID of the vServer.

Type: *xs:string*

Examples

Example 1

The following example creates and starts a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=RunVserver&Version=1&Timestamp=1321379615665&Expires=1321379915665&vserverType=457&keyName=myKeyPair&vnets.1=VNET-84ada392-1c13-4f86-8365-1cf7f9c8aadf&name=myVserver&ipAddresses.1=192.168.0.2&serverTemplateId=TMPL-9e4a9ed3-e675-45f1-9d7c-b21c25a55632&AccessKeyId=AK_
```

1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result
xmlns="http://www.oracle.com/xml/ns/iaas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RunVserverResult"
requestId="120"><vserverId>VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6</vserverId></
result>
```

See Also

- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)
- [RunVserversRequest](#)
- [SendMessageToVserverRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

RunVserversRequest

Description

Action to create and start multiple vServers.

Request Parameters

name

Name of the vServers.

Type: *GenericString*

Required: *Yes*

description

Description of the vServers.

Type: *GenericString*

Required: *No*

serverTemplateId

ID of the server template.

Type: *ServerTemplateIdType*

Required: *Yes*

num

Number of vServers.

Type: *PositiveInteger*

Default: *1*

Required: *No*

vnets

List of one or more virtual network IDs.

Type: *VnetIdType*

Required: *Yes*

keyName

Name of the key pair.

Type: *GenericString*

Required: *No*

vserverType

ID of the vServer type.

Type: *GenericString*

Required: *Yes*

volumes

List of one or more volume IDs.

Type: *VolumeIdType*

Required: *No*

distGroup

ID of the distribution group.

Type: *DistributionGroupIdType*

Required: *No*

ha

Flag to indicate if high availability is enabled for the vServers. The flag is boolean; possible values are true or false.

Type: *GenericString*

Required: *No*

messages

List of messages sent as key-value pairs to the vServers.

Type: *Message*

Required: *No*

hostname

A host name to set or override the host name for the vServers. It must be RFC 1123 compliant.

Type: *HostnameType*

Required: *No*

rootPassword

A password to set or override the root password for the vServers.

Type: *GenericString*

Required: *No*

Result Elements

The result elements are contained in `RunVserversResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

vserverIds

List of one or more vServer IDs.

Type: *xs:string*

Examples

Example 1

The following example creates and starts two vServers.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=RunVservers&Version=1&Timestamp=1320343596868&Expires=1320343896868&vserverType=457&num=2&keyName=myKeyPair&vnet
```

```
s.1=VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4&name=myVserver&serverTemplateId=TMPL-2f313208-433c-4b92-aae6-6373c38b795e&AccessKeyId=AK_32&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RunVserversResult" requestId="409"><vserverIds>VSRV-d6800889-f59b-4798-a57d-3f9f31b0cf1c</vserverIds><vserverIds>VSRV-d6500889-f59b-4567-a65g-3f9f31b0se1d</vserverIds></result>
```

See Also

- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)
- [RunVserverRequest](#)
- [SendMessagesToVserverRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

SendMessagesToVserverRequest

Description

Action to send a list of messages to a vServer. These messages can be read from the guest operating system.

Request Parameters

vserverId

ID of the vServer.

Type: *VserverIdType*

Required: *Yes*

messages

List of one or more message key names and values.

Type: *Message*

Required: *Yes*

Result Elements

The result elements are contained in `SendMessagesToVserverResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example sends a message to a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=SendMessagesToVserver&Version=1&Timestamp=1321381585564&Expires=1321381885564&messages.1.value=Running&messages.1.key=myStatus&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="SendMessagesToVserverResult" requestId="123"/>
```

See Also

- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)
- [RunVserverRequest](#)

- [RunVserversRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

StartVserversRequest

Description

Action to start one or more stopped or shutdown vServers.

Request Parameters

vserverIds

One or more vServer IDs.

Type: *VserverIdType*

Required: *Yes*

Result Elements

The result elements are contained in StartVserversResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example starts a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=StartVservers&Version=1&Timestamp=1320104759496&Expires=1320105059496&vserverIds.1=VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="StartVserversResult" requestId="275"/>
```

See Also

- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)
- [SendMessageToVserverRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

StopVserversRequest

Description

Action to stop one or more running vServers.

Request Parameters

vserverIds

List of one or more vServer IDs.

Type: *VserverIdType*

Required: *Yes*

force

Flag to force the action.

Type: *xs:boolean*

Default: *0 (false)*

Required: *No*

Result Elements

The result elements are contained in StopVserversResult.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example stops a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=StopVservers&Version=1&Timestamp=1320105610783&Expires=1320105910783&vserverIds.1=VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="StopVserversResult" requestId="278"/>
```

See Also

- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)
- [RunVserverRequest](#)

- [RunVserversRequest](#)
- [SendMessageToVserverRequest](#)
- [StopVserversRequest](#)
- [TerminateVserversRequest](#)

TerminateVserversRequest

Description

Action to delete one or more vServers.

Request Parameters

vserverIds

List of one or more vServer IDs.

Type: *VserverIdType*

Required: *Yes*

force

Flag to force the action.

Type: *xs:boolean*

Default: *0 (false)*

Required: *No*

Result Elements

The result elements are contained in `TerminateVserversResult`.

requestId

ID of the request.

Type: *xs:string*

Required: *Yes*

Examples

Example 1

The following example deletes a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=TerminateVservers&Version=1&Timestamp=1321385662260&Expires=1321385962260&vserverIds.1=VSRV-0fb57293-347c-4717-96ef-6dd23154596f&force=true&AccessKeyId=AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

HTTP Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><result xmlns="http://www.oracle.com/xml/ns/iaas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="TerminateVserversResult" requestId="160"/>
```

See Also

- [RebootVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)

- [RunVserverRequest](#)
- [RunVserversRequest](#)
- [SendMessageToVserverRequest](#)
[StartVserversRequest](#)
- [StopVserversRequest](#)

Data Types

The following data types are described in this section:

- [AccountIdType](#)
- [DescribeAccessKeysResultItem](#)
- [DescribeAccountsResultItem](#)
- [DescribeDistributionGroupsResultItem](#)
- [DescribeIpAddressesResultItem](#)
- [DescribeKeyPairsResultItem](#)
- [DescribeServerTemplatesResultItem](#)
- [DescribeSnapshotsResultItem](#)
- [DescribeTagsResultItem](#)
- [DescribeVdcCapabilitiesResultItem](#)
- [DescribeVnetsResultItem](#)
- [DescribeVolumesResultItem](#)
- [DescribeVserverMetricsResultItem](#)
- [DescribeVserverTypesResultItem](#)
- [DistributionGroupIdType](#)
- [FilterItem](#)
- [GenericString](#)
- [Message](#)
- [PositiveInteger](#)
- [ResourceAttribute](#)
- [ResourceIdType](#)
- [SnapshotIdType](#)
- [ServerTemplateIdType](#)
- [TagType](#)
- [URLType](#)
- [VnetIdType](#)
- [VolumeIdType](#)
- [VserverIdType](#)

AccountIdType

Data Type Elements

This data type defines the pattern value of an account ID.

Type: *xs:string*

Pattern value: `ACC-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Ancestors

None

Actions

- [CreateAccessKeyAsObjectRequest](#)
- [CreateAccessKeyToFileRequest](#)
- [DescribeAccountsRequest](#)
- [RegisterAccessKeyRequest](#)

DescribeAccessKeysResultItem

Data Type Elements

accessKeyID

ID of the access key.

Type: *xs:string*

account

Name of the account.

Type: *xs:string*

Ancestors

None

Actions

[DescribeAccessKeysRequest](#)

DescribeAccountsResultItem

Data Type Elements

account

Account ID.

Type: *xs:string*

name

Name of the account.

Type: *xs:string*

description

Description of the account.

Type: *xs:string*

Ancestors

None

Actions

[DescribeAccountsRequest](#)

DescribeDistributionGroupsResultItem

Data Type Elements

id

ID of the distribution group.

Type: *xs:string*

name

Name of the distribution group.

Type: *xs:string*

description

Description of the distribution group.

Type: *xs:string*

status

Status of the distribution group.

Type: *xs:string*

tags

List of tags associated with the distribution group.

Type: *TagType*

vservers

List of vServer IDs associated with the distribution group.

Type: *VserverIdType*

size

Size of the distribution group.

Type: *xs:int*

Ancestors

None

Actions

[DescribeDistributionGroupsRequest](#)

DescribeIpAddressesResultItem

Data Type Elements

ipAddress

Reserved IP address.

Type: *xs:string*

vnet

ID of the virtual network.

Type: *VnetIdType*

vserver

ID of the vServer associated with the IP address.

Type: *VserverIdType*

Ancestors

None

Actions

[DescribeIpAddressesRequest](#)

DescribeKeyPairsResultItem

Data Type Elements

name

Name of the key pair.

Type: *xs:string*

fingerprint

Fingerprint of the key pair.

Type: *xs:string*

Ancestors

None

Actions

[DescribeKeyPairsRequest](#)

DescribeServerTemplatesResultItem

Data Type Elements

id

ID of the server template.

Type: *xs:string*

name

Name of the server template.

Type: *xs:string*

description

Description of the server template.

Type: *xs:string*

status

Status of the server template.

Type: *xs:string*

tags

List of none or more tags for the server template.

Type: *TagType*

size

Size of the server template in GB.

Type: *xs:long*

public

Flag to indicate if the server template is available for other accounts.

Type: *xs:boolean*

imageType

Type of the image, VM template or assembly.

Type: *GenericString*

readOnly

Flag to indicate if the server template is read only.

Type: *xs:boolean*

Ancestors

None

Actions

[DescribeServerTemplatesRequest](#)

DescribeSnapshotsResultItem

Data Type Elements

id

ID of the snapshot.

Type: *xs:string*

name

Name of the snapshot.

Type: *xs:string*

description

Description of the snapshot.

Type: *xs:string*

status

Status of the snapshot.

Type: *xs:string*

tags

List of none or more tags for the snapshot.

Type: *TagType*

Ancestors

None

Actions

[DescribeSnapshotsRequest](#)

DescribeTagsResultItem

Data Type Elements

resourceId

ID of the resource.

Type: *xs:string*

name

Name of the tag.

Type: *xs:string*

value

Value of the tag.

Type: *xs:string*

Ancestors

None

Actions

[DescribeTagsRequest](#)

DescribeVdcCapabilitiesResultItem

Data Type Elements

name

Name of vDC capability.

Type: *GenericString*

value

Value of the vDC capability.

Type: *GenericString*

Ancestors

None

Actions

[DescribeVdcCapabilitiesRequest](#)

DescribeVnetsResultItem

Data Type Elements

id

ID of the virtual network.

Type: *xs:string*

name

Name of the virtual network.

Type: *xs:string*

description

Description of the virtual network.

Type: *xs:string*

status

Status of the virtual network.

Type: *xs:string*

tags

List of none or more tags for the virtual network.

Type: *TagType*

ipAddress

Subnet of the virtual network.

Type: *GenericString*

Ancestors

None

Actions

[DeleteVnetRequest](#)

DescribeVolumesResultItem

Data Type Elements

id

ID of the volume.

Type: *xs:string*

name

Name of the volume.

Type: *xs:string*

description

Description of the volume.

Type: *xs:string*

status

Status of the volume.

Type: *xs:string*

tags

List of none or more tags for the volume.

Type: *TagType*

size

Size of the volume in GB.

Type: *xs:double*

shared

Flag to indicate if the volume is shared.

Type: *xs:boolean*

vServers

List of vServer IDs associated with the volume.

Type: *VserverIdType*

Ancestors

None

Actions

[DescribeVolumesRequest](#)

DescribeVserverMetricsResultItem

Data Type Elements

id

ID of the vServer.

Type: *xs:string*

name

Name of the vServer.

Type: *xs:string*

description

Description of the vServer.

Type: *xs:string*

status

Status of the vServer.

Type: *xs:string*

tags

List of tags associated with the vServer.

Type: *TagType*

cpuUsage

vServer cpu usage.

Type: *xs:float*

Ancestors

None

Actions

[DescribeVserverMetricsRequest](#)

DescribeVserversResultItem

Data Type Elements

id

ID of the vServer.

Type: *xs:string*

name

Name of the vServer.

Type: *xs:string*

description

Description of the vServer.

Type: *xs:string*

status

Status of the vServer.

Type: *xs:string*

tags

List of tags associated with the vServer.

Type: *TagType*

vnets

List of virtual network IDs to which the vServer belongs to.

Type: *VnetIdType*

ipAddresses

List of IP addresses associated with the vServer.

Type: *xs:string*

serverTemplateId

ID of the server template associated with the vServer.

Type: *ServerTemplateIdType*

keyName

Name of the key pair associated with the vServer.

Type: *GenericString*

vserverType

ID of the vServer type associated with the vServer.

Type: *GenericString*

ha

Flag to indicate whether automatic recovery is enabled for the vServer. Flag is boolean, possible values are true or false.

Type: *GenericString*

distributionGroup

ID of the distribution group associated with the vServer.

Type: *GenericString*

volumes

List of volume IDs associated with the vServer.

Type: *xs:string*

vcpu

Number of vCPUs of the vServer.

Type: *xs:integer*

memoryMb

Total memory allocated for the vServer.

Type: *xs:long*

dedicatedStorageMb

Total storage dedicated to the vServer.

Type: *xs:long*

attachedStorageMb

Total storage of the volumes attached to the vServer.

Type: *xs:long*

Ancestors

None

Actions

[DescribeVserversRequest](#)

DescribeVserverTypesResultItem

Data Type Elements

id

ID of the vServer type.

Type: *xs:string*

name

Name of the vServer type.

Type: *xs:string*

description

Description of the vServer type.

Type: *xs:string*

status

Status of the vServer type.

Type: *xs:string*

tags

List of none or more tags of the vServer type.

Type: *TagType*

memorySize

Memory size of the vServer type in GB.

Type: *xs:long*

storageSize

Storage size of the vServer type in GB.

Type: *xs:long*

vcpu

Number of virtual CPUs of the vServer type.

Type: *xs:integer*

Ancestors

None

Actions

[DescribeVserverTypesRequest](#)

DistributionGroupIDType

Data Type Elements

This data type defines the pattern value of a distribution group ID.

Type: *xs:string*

Pattern value: *DG-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}*

Ancestors

None

Actions

- [DeleteDistributionGroupRequest](#)
- [DescribeDistributionGroupsRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)

FilterItem

Data Type Elements

filterName

Name of the filter. The name of an attribute for specified resource.

Type: *GenericString*

filterValue

Value of the filter. The value for the specified resource's attribute.

Type: *GenericString*

Ancestors

None

Actions

- [DescribeIpAddressesRequest](#)
- [DescribeKeyPairsRequest](#)
- [DescribeServerTemplatesRequest](#)
- [DescribeSnapshotsRequest](#)
- [DescribeTagsRequest](#)
- [DescribeVdcCapabilitiesRequest](#)
- [DescribeVnetsRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeVserversRequest](#)

GenericString

Data Type Elements

Simple data type with the following definition.

Type: *xs:string*

maxLength value: 256

enumeration value: 200

Ancestors

- [FilterItem](#)
- [Message](#)
- [TagType](#)

Actions

- [CreateAccessKeyAsObjectRequest](#)
- [CreateAccessKeyToFileRequest](#)
- [CreateKeyPairAsObjectRequest](#)
- [CreateKeyPairToFileRequest](#)
- [DescribeAccessKeysRequest](#)
- [DescribeAccountsRequest](#)
- [DescribeKeyPairsRequest](#)
- [ImportKeyPairRequest](#)
- [RegisterAccessKeyRequest](#)
- [RegisterKeyPairRequest](#)

HostnameType

Data Type Elements

Simple data type for a host name with the following definition.

Type: *xs:string*

pattern value: `[A-Za-z\\d] | ([A-Za-z\\d][A-Za-z\\d\\-]{0,253}[A-Za-z\\d])`

Ancestors

None

Actions

- [RunVserverRequest](#)
- [RunVserversRequest](#)

Message

Data Type Elements

key

Key associated with the message.

Type: *GenericString*

value

The value of the message.

Type: *GenericString*

Ancestors

None

Actions

[SendMessageToVserverRequest](#)

PositiveInteger

Data Type Elements

This data type defines the pattern value of a PositiveInteger data type.

Type: *xs:integer*

minInclusive value: *1*

Ancestors

None

Actions

- [AllocateIpAddressesRequest](#)
- [CreateVnetRequest](#)
- [CreateVolumeRequest](#)
- [RunVserversRequest](#)

ResourceAttribute

Data Type Elements

name

Name of an attribute for a resource.

Type: *xs:string*

value

Value for the specified resource's attribute.

Type: *xs:string*

Ancestors

None

Actions

- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

ResourceIdType

Data Type Elements

This data type defines the pattern value of a resource ID.

Type: *xs:string*

Pattern value: `[A-Z]{1,4}-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Ancestors

None

Actions

- [CreateTagsRequest](#)
- [DeleteTagsRequest](#)
- [DescribeAttributesRequest](#)
- [ModifyAttributesRequest](#)

SnapshotIdType

Data Type Elements

This data type defines the pattern value of a snapshot ID.

Type: *xs:string*

Pattern value: *SNAP-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}*

Ancestors

None

Actions

- [CreateSnapshotRequest](#)
- [CreateVolumeRequest](#)
- [DeleteSnapshotRequest](#)

ServerTemplateIdType

Data Type Elements

This data type defines the pattern value of a server template ID.

Type: *xs:string*

Pattern value: *TMPL-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}*

Ancestors

[DescribeServerTemplatesRequest](#)

Actions

- [DeregisterServerTemplateRequest](#)
- [DescribeServerTemplatesRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)

TagType

Data Type Elements

name

The name of the tag.

Type: *GenericString*

value

The value of the tag.

Type: *GenericString*

Ancestors

- [DescribeServerTemplatesResultItem](#)
- [DescribeSnapshotsRequest](#)
- [DescribeVnetsResultItem](#)
- [DescribeVolumesResultItem](#)
- [DescribeVserverMetricsResultItem](#)
- [DescribeVserverTypesResultItem](#)

Actions

- [CreateTagsRequest](#)
- [DeleteTagsRequest](#)
- [DescribeServerTemplatesRequest](#)
- [DescribeSnapshotsRequest](#)
- [DescribeVnetsRequest](#)
- [DescribeVolumesRequest](#)
- [DescribeVserversRequest](#)
- [DescribeVserverTypesRequest](#)

URLType

Data Type Elements

This data type defines the pattern value of a URL.

Type: *xs:string*

Ancestors

None

Actions

- [ImportVolumeRequest](#)
- [RegisterServerTemplateFromUrlRequest](#)
- [RegisterServerTemplatesFromAssemblyRequest](#)

VnetIdType

Data Type Elements

This data type defines the pattern value of a virtual network ID.

Type: *xs:string*

Pattern value: *VNET-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}*

Ancestors

- [DescribeIpAddressesResultItem](#)
- [DescribeVserverMetricsResultItem](#)

Actions

- [AllocateIpAddressesRequest](#)
- [DeleteVnetRequest](#)
- [DescribeIpAddressesRequest](#)
- [DescribeVnetsRequest](#)
- [DescribeVserversRequest](#)
- [ReleaseIpAddressesRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)

VolumeldType

Data Type Elements

This data type defines the pattern value of a volume ID.

Type: *xs:string*

Pattern value: *VOL-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}*

Ancestors

None

Actions

- [AttachVolumesToVserverRequest](#)
- [CreateSnapshotRequest](#)
- [DeleteVolumeRequest](#)
- [DescribeVolumesRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [RunVserverRequest](#)
- [RunVserversRequest](#)

VserverIdType

Data Type Elements

This data type defines the pattern value of a vServer ID.

Type: *xs:string*

Pattern value: *VSRV-[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}*

Ancestors

[DescribeIpAddressesRequest](#)

Actions

- [AttachVolumesToVserverRequest](#)
- [DescribeIpAddressesRequest](#)
- [DescribeVserversRequest](#)
- [DetachVolumesFromVserverRequest](#)
- [RebootVserversRequest](#)
- [RegisterServerTemplateFromVserverRequest](#)
- [SendMessageToVserverRequest](#)
- [StartVserversRequest](#)
- [StopVserversRequest](#)
- [ReceiveMessageFromVserverRequest](#)
- [TerminateVserversRequest](#)

Cloud Infrastructure CLI Reference

This chapter provides a reference to using the cloud infrastructure CLI once it is installed. The chapter includes a description, syntax, options, and examples for each command.

The following topics are covered in this chapter:

- [Common Command Options](#)
- [List of Commands](#)

Common Command Options

All the cloud infrastructure CLI commands have one of two common prefixes:

- `akm` – Use only for commands related to access keys, for example `akm-create-access-key`.
- `iaas` – Use for all other cloud computing management commands, for example `iaas-create-volume`.

All the cloud infrastructure CLI commands share some common options. Other options are common for commands with the same prefix. [Table 5–1](#) lists the command options and their descriptions, and indicates whether the option is required for the command.

Table 5–1 Common Command Options

Command Type	Option	Description	Required
akm and iaas	<code>--base-url <base_url></code>	Base URL of the Enterprise Controller where the Web service is running. <code>https://<EnterpriseController host></code> .	Yes
akm	<code>--user</code>	The user name for authentication.	Yes
akm	<code>--password-file -p <pw_file></code>	Path to the password file for authentication. If this option is not supplied, then the cloud user is prompted for a password.	No
iaas	<code>--access-key-file -a <access_key_file></code>	Path of the file storing the access key	Yes
akm and iaas	<code>--help -h</code>	Explains the command usage and its argument.	No

Table 5–1 (Cont.) Common Command Options

Command Type	Option	Description	Required
akm and iaas	--header -H	Adds a header row to the output	No
akm and iaas	--sep <separator>	Specifies a column separator character. The default separator is <i>TAB</i> .	No
akm and iaas	--xml	Displays the output result in XML format. The default output is in table format.	No
akm and iaas	--verbose -v	Starts the command in verbose mode	No
akm and iaas	--debug -D	Starts the command in debug mode	No

Alternatively, the value for some common required options can be specified using the environment variables: `IAAS_PASSWORD_FILE`, `IAAS_BASE_URL`, `IAAS_USER`, and `IAAS_ACCESS_KEY_FILE`.

List of Commands

Each command is delivered as separate file. Every command has a short and long form. The following CLI commands are described in this section:

Functionality	Commands
Account access	<code>akm-create-access-key</code> , <code>akm-delete-access-key</code> , <code>akm-describe-access-keys</code> , <code>akm-describe-accounts</code>
Server template management	<code>iaas-create-server-template-from-assembly</code> , <code>iaas-create-server-template-from-url</code> , <code>iaas-create-server-template-from-vserver</code> , <code>iaas-delete-server-template</code> , <code>iaas-describe-server-templates</code>
Virtual network management	<code>iaas-create-vnet</code> , <code>iaas-delete-vnet</code> , <code>iaas-describe-vnets</code>
vIP address management	<code>iaas-allocate-ip-addresses</code> , <code>iaas-describe-ip-addresses</code> , <code>iaas-release-ip-addresses</code>
Volume management	<code>iaas-attach-volumes-to-vserver</code> , <code>iaas-create-volume</code> , <code>iaas-delete-volume</code> , <code>iaas-describe-volumes</code> , <code>iaas-detach-volumes-from-vserver</code> , <code>iaas-import-volume</code>
Snapshot management	<code>iaas-create-snapshot</code> , <code>iaas-delete-snapshot</code> , <code>iaas-describe-snapshots</code>
Key pair management	<code>iaas-create-key-pair</code> , <code>iaas-delete-key-pair</code> , <code>iaas-describe-key-pairs</code> , <code>iaas-import-key-pair</code>
vServer management	<code>iaas-describe-vservers</code> , <code>iaas-describe-vserver-types</code> , <code>iaas-reboot-vservers</code> , <code>iaas-receive-message-from-vserver</code> , <code>iaas-run-vserver</code> , <code>iaas-run-vservers</code> , <code>iaas-send-messages-to-vserver</code> , <code>iaas-start-vservers</code> , <code>iaas-stop-vservers</code> , <code>iaas-terminate-vservers</code>
Distribution group management	<code>iaas-create-distribution-group</code> , <code>iaas-delete-distribution-group</code> , <code>iaas-describe-distribution-groups</code>
Resource attribute management	<code>iaas-describe-attributes</code> , <code>iaas-modify-attributes</code>
Tag management	<code>iaas-create-tags</code> , <code>iaas-delete-tags</code> , <code>iaas-describe-tags</code>
vDC capabilities	<code>iaas-describe-vdc-capabilities</code>

akm-create-access-key

Creates an access key for a cloud user.

Syntax

```
akm-create-access-key|akm-cak [--base-url <base_url>] [--debug] [--header]
                               [--user <username>] [--help]
                               [--password-file <password_file>]
                               [--sep <separator>][--xml]
                               [--trust-store <truststore_file>]
                               [--for-user <username>]
                               --account <account_id> [--verbose]
                               --access-key-file <access_key_file>
```

Description

This command delivers the access key ID and the access key file containing the private key of the newly created access key for the specified cloud user. The command also registers the access key to the specified account.

The access key file created by this command is used for authentication of every `iaas` command.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--user <username>

User name of the cloud user. The command validates the cloud user in the Oracle Enterprise Manager Ops Center instance and verifies that the cloud user has the correct privileges for the specified account. If this option is not specified, then the user name is taken from the value set in the `IAAS_USER` environment variable.

--help | -h

Displays the usage information for this command.

--access-key-file | -a <access_key_file>

Complete path and name of the file that will store the access key created after using this command (for example, `/tmp/ak.file`). If a path is not specified, the file is created in the current directory.

--password-file | -p <password_file>

Complete path and name of the file containing the password of the cloud user (for example, `/tmp/akpasswd.file`). If a path is not specified for this file, then the command-line interface looks for the file in the current directory. If this option is not

specified, and the value is not set in the `IAAS_PASSWORD_FILE` environment variable, then the cloud user is prompted for a password.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file that stores the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--for-user | -f <username>

User name of another cloud user to create an access key on behalf of that cloud user. The user executing the command using this option requires cloud administrator privileges.

--account | -d <account_id>

Account ID of the account for which the access key is created. The command checks that the specified user is already registered for that account. To get the account ID, use the [akm-describe-accounts](#) command.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates an access key with the ID `AK_3`. It uses the following options and values:

Option	Option Value
<code>base-url</code>	<code>https://<EnterpriseControllerHostname>/</code>
<code>user</code>	<code><username></code>
<code>password-file</code>	<code>~/pwd.file</code>
<code>account</code>	<code>ACC-e7ed508a-dad3-4866-a41a-bf8850163b2e</code>
<code>access-key-file</code>	<code>ak.file</code>

```
bash-3.2$ ./akm-create-access-key --base-url
https://<EnterpriseControllerHostname>/ --user <username> --password-file
~/pwd.file --account ACC-e7ed508a-dad3-4866-a41a-bf8850163b2e -a ak.file
```

```
AK_3
bash-3.2$
```

Example 2

This example creates an access key with ID `AK_4` using the value set in the environment variables. It uses the following options and values:

Option	Environment Variable	Value
<code>base-url</code>	<code>IAAS_BASE_URL</code>	<code>https://<EnterpriseControllerHostname>/</code>

Option	Environment Variable	Value
user	IAAS_USER	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file
account	NA	ACC-g7fg508a-dad3-4866-a41a-bf8850163c3f
access-key-file	NA	ak.file

```
bash-3.2$ ./akm-create-access-key --account
ACC-g7fg508a-dad3-4866-a41a-bf8850163c3f --access-key-file ak.file
```

```
AK_4
bash-3.2$
```

Example 3

In this example a cloud administrator creates, on behalf of a cloud user, an access key with ID *AK_10*. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
user	IAAS_USER	cloudadmin
for-user	NA	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file
account	NA	ACC-g7fg508a-dad3-4866-a41a-bf8850163c3f
access-key-file	NA	ak.file

```
bash-3.2$ ./akm-create-access-key --account
ACC-g7fg508a-dad3-4866-a41a-bf8850163c3f --access-key-file ak.file --for-user
<username>
```

```
AK_10
bash-3.2$
```

Exit Status

(Success) – The command returns the access key ID of the newly created access key. When the [akm-describe-access-keys](#) command is used, the new access key ID is listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [akm-delete-access-key](#)
- [akm-describe-access-keys](#)
- [akm-describe-accounts](#)

akm-delete-access-key

Unregisters an access key from the specified account.

Syntax

```
akm-delete-access-key|akm-delak  [--base-url <base_url>] [--debug] [--header]
                                  [--user <user name>] [--help]
                                  [--password-file <pw_file>]
                                  [--sep <separator>]
                                  [--trust-store <truststore_file>]
                                  [--verbose] [--xml]
                                  <access_key_id>
```

Description

This command deletes the access key association to the account. The access key file is not deleted with this command, but the access key is no longer usable.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--user <username>

User name of the cloud user. The command validates the cloud user in the Oracle Enterprise Manager Ops Center instance and verifies that the cloud user has the correct privileges for the specified account. If this option is not specified, then the user name is taken from the value set in the `IAAS_USER` environment variable.

--help | -h

Displays the usage information for this command.

--password-file | -p <password_file>

Name of the file containing the password of the cloud user. If a path is not specified for this file, then the command-line interface looks for the file in the current directory. If this option is not specified, and the value is not set in the `IAAS_PASSWORD_FILE` environment variable, then the cloud user is prompted for a password.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file that stores the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

<access_key_id>

Access key ID of the access key that will be unregistered. Access key IDs are displayed in the user interface or by using the [akm-describe-access-keys](#) command.

Examples

Example 1

This example unregisters an access key. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
user	<username>
password-file	~/pwd.file
access-key-id	AK_3

```
bash-3.2$ ./akm-delete-access-key --base-url
https://<EnterpriseControllerHostname>/ --user <username> --password-file
~/pwd.file
```

```
AK_3
bash-3.2$
```

Example 2

This example unregisters an access key. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
user	IAAS_USER	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file
access-key-id	NA	AK_4

```
bash-3.2$ ./akm-delete-access-key AK_4

bash-3.2$
```

Exit Status

(Success) – The command does not return any value, only the command prompt is returned. When the [akm-describe-access-keys](#) command is used, the access key is not listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [akm-create-access-key](#)

- [akm-describe-access-keys](#)
- [akm-describe-accounts](#)

akm-describe-access-keys

Displays a list of the access keys of a cloud user.

Syntax

```
akm-describe-access-keys|akm-dak  [--base-url <base_url>] [--debug] [--header]
                                   [--user <user name>] [--help]
                                   [--password-file <password_file>]
                                   [--sep <separator>]
                                   [--trust-store <truststore_file>]
                                   [--for-user <username>]
                                   [--verbose] [--xml]
```

Description

This command list information about the access keys created for a cloud user. The command displays the access key ID, the account ID, and user name of the cloud user for each access key available for the specified cloud user.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--user <username>

Username of the cloud user. The command validates the cloud user in the Oracle Enterprise Manager Ops Center instance and verifies that the cloud user has the correct privileges for the specified account. If this option is not specified, then the user name is taken from the value set in the `IAAS_USER` environment variable.

--help | -h

Displays the usage information for this command.

--password-file | -p <password_file>

Name of the file containing the password of the cloud user. If a path is not specified for this file, then the command-line interface looks for the file in the current directory. If this option is not specified, and the value is not set in the `IAAS_PASSWORD_FILE` environment variable, then the cloud user is prompted for a password.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file that stores the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--for-user | -f <username>

User name of a different cloud user to display the access keys created for that cloud user. The cloud user using this option requires cloud administrator privileges.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example displays the access keys of a cloud user. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
user	<username>
password-file	~/pwd.file

```
bash-3.2$ ./akm-describe-access-keys --base-url
https://<EnterpriseControllerHostname>/ --user <username> --password-file
~/pwd.file
```

```
AK_3    ACC-d7td508a-dad3-4866-a41a-bf8850163c3g    <username>
bash-3.2$
```

Example 2

This example displays the access keys of a cloud user, adding a header row to the output. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
user	IAAS_USER	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file

```
bash-3.2$ ./akm-describe-access-keys -H
accessKeyId    account    user
```

```
AK_3    ACC-d7td508a-dad3-4866-a41a-bf8850163c3g    <username>
bash-3.2$
```

Exit Status

(Success) – The command displays a list containing the access key ID, the account ID, and username of the specified cloud user.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [akm-create-access-key](#)

- [akm-delete-access-key](#)
- [akm-describe-accounts](#)

akm-describe-accounts

Displays account information for a cloud user.

Syntax

```
akm-describe-accounts|akm-dac  [--base-url <base_url>] [--debug] [--header]
                                [--user <user name>] [--help]
                                [--password-file <password_file>]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--for-user <username>]
                                [--account <account_id>] [--verbose]
                                [--xml]
```

Description

This command lists information about the accounts available for a cloud user. The command displays the user name of the specified cloud user along with the account ID, name, and description of each account available for that cloud user.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--user <username>

Username of the cloud user. The command validates the cloud user in the Oracle Enterprise Manager Ops Center instance and verifies that the cloud user has the correct privileges for the specified account. If this option is not specified, then the user name is taken from the value set in the `IAAS_USER` environment variable.

--help | -h

Displays the usage information for this command.

--password-file | -p <password_file>

Name of the file containing the password of the cloud user. If a path is not specified for this file, then the command-line interface looks for the file in the current directory. If this option is not specified, and the value is not set in the `IAAS_PASSWORD_FILE` environment variable, then the cloud user is prompted for a password.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--for-user | -f <username>

User name of another cloud user. This option validates that the cloud user using this option has cloud administrator privileges.

--account | -d <account_id>

Account ID of the account. This option can be used as a filter to limit the number of accounts displayed.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example displays the account information for a cloud user. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
user	<username>
password-file	~/pwd.file

```
bash-3.2$ ./akm-describe-accounts --base-url
https://<EnterpriseControllerHostname>/ --user <username> --password-file
~/pwd.file

ACC-f7gd508a-dad3-4866-a41a-bf8850163c3g      account0      account0 for user
<username>          <username>
ACC-95gdb073-d569-4a45-ac24-f14eba689a8g      account1      account1 for user
<username>          <username>
ACC-eag64db9-c0ee-43de-bb17-55cfb9d105fg      account2      account2 for user
<username>          <username>
bash-3.2$
```

Example 2

This example displays the account information for a different cloud user, and includes a header row. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
user	IAAS_USER	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file
for-user	NA	<username2>

```
bash-3.2$ ./akm-describe-accounts --for-user <username2> -H

account name      description      user
ACC-f7gd508a-dad3-4866-a41a-bf8850163c3g      account0      account0 for user
<username2>          <username>
```

```
ACC-95gdb073-d569-4a45-ac24-f14eba689a8g      account1      account1 for user  
<username2>          <username>  
bash-3.2$
```

Exit Status

(Success) – The command returns a list containing the account ID, the account name, the account description, and user name of the specified cloud user.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [akm-create-access-key](#)
- [akm-delete-access-key](#)
- [akm-describe-access-keys](#)

iaas-allocate-ip-addresses

Allocates a number of IP addresses from a virtual network.

Syntax

```
iaas-allocate-ip-addresses|iaas-aip [--base-url <base_url>] [--debug]
                                     [--header] [--help]
                                     [--access-key-file <access_key_file>]
                                     [--vnet] <vnet-id>
                                     [--num <num>]
                                     [--sep <separator>]
                                     [--trust-store <truststore_file>]
                                     [--verbose] [--xml]
```

Description

This command allocates a number of IP addresses from a public or private virtual network that can be associated later with a vServer.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vnet <vnet-id>

vNet ID from which IP addresses are allocated.

--num <num>

Number of IP addresses to be allocated. The default value is 1.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example allocates an IP address from the specified virtual network. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vNet ID	VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c

```
bash-3.2$ ./iaas-allocate-ip-addresses --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --vnet
VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c
```

```
192.168.0.1
bash-3.2$
```

Example 2

This example allocates three IP addresses from the specified virtual network. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vNet ID	VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c

```
bash-3.2$ ./iaas-allocate-ip-addresses --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --vnet
VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c --num 3
```

```
192.168.0.2
192.168.0.3
192.168.0.4
bash-3.2$
```

Exit Status

(Success) – The command returns a list of the allocated IP addresses. When using the [iaas-describe-ip-addresses](#) command, the new allocated IP addresses are listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-vnet](#)

- [iaas-describe-ip-addresses](#)
- [iaas-describe-vnets](#)
- [iaas-release-ip-addresses](#)

iaas-attach-volumes-to-vserver

Attaches one or more volumes to a vServer.

Syntax

```
iaas-attach-volumes-to-vserver|iaas-attvol [--base-url <base_url>] [--debug]
                                           [--access-key-file <access_key_file>]
                                           [--debug] [--header][--help]]
                                           --vserver-id <vserver-id>
                                           [--volume-ids <vol-id>[,<vol-id>]*
                                           [--sep <separator>]
                                           [--trust-store <truststore_file>]
                                           [--verbose] [--xml]
```

Description

This command attaches one or more volumes to a vServer.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vserver-id | -i <vserver-id>

ID of the vServer to which the volume will be attached.

--volume-ids <vol-id>[,<vol-id>]*

IDs of the volumes to be attached.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example attaches a volume to a vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vserver-id	VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6

```
bash-3.2$ ./iaas-attach-volumes-to-vserver --base-url
https://<EnterpriseControllerHostname>/ -a ~/ak.file -vserver-id
VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6 --volume-ids
VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15

bash-3.2$
```

Exit Status

(Success) – The command attaches the specified volume to the vServer. This command does not return any value, only the command prompt is returned. When the [iaas-describe-volumes](#) command is used, the vServer is listed as part of the volume information.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-volume](#)
- [iaas-describe-volumes](#)
- [iaas-describe-vservers](#)
- [iaas-detach-volumes-from-vserver](#)
- [iaas-import-volume](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-create-distribution-group

Creates a new distribution group for an account.

Syntax

```
iaas-create-distribution-group|iaas-cdg [--base-url <base_url>] [--debug]
                                     [--desc <descr>][--header] [--help]
                                     [--access-key-file <access_key_file>]
                                     --name <name>
                                     [--sep <separator>]
                                     [--size <size>]
                                     [--trust-store <truststore_file>]
                                     [--verbose] [--xml]
```

Description

This command creates a distribution group for an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--desc | -d <descr>

An optional description for the distribution group.

--name | -n <name>

Name of the distribution group.

--sep <separator>

Column separator character. The default separator is *TAB*.

--size <size>

Size of the distribution group. The size is the maximum limit of vServers that can be added to a distribution group. The default size is 50000.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates a distribution group for an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myDistributionGroup

```
bash-3.2$ ./iaas-create-distribution-group --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --name myDistributionGroup

DG-068ae84c-d0fc-406d-aa37-0be4f88d411c
bash-3.2$
```

Example 2

This example creates a distribution group with size for 10 vServers. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myDistributionGroup2

```
bash-3.2$ ./iaas-create-distribution-group --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --name myDistributionGroup2
--size 10

DG-9eb288c2-85e7-4392-80a6-d1c1709de4cd
bash-3.2$
```

Exit Status

(Success) – The command returns the distribution group ID. When using the [iaas-describe-distribution-groups](#) command, the new distribution group is listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-delete-distribution-group](#)

- [iaas-describe-distribution-groups](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-create-key-pair

Creates a key pair for an account.

Syntax

```
iaas-create-key-pair|iaas-ckp [--base-url <base_url>] [--debug]
                               [--header] [--help]
                               [--access-key-file <access_key_file>]
                               --key-file <key-file-name>
                               --key-name <key-pair-name>
                               [--sep <separator>]
                               [--trust-store <truststore_file>]
                               [--verbose] [--xml]
```

Description

This command creates a key pair for an account. The command stores the private key of the key pair in the specified key file.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--key-name | -n <key-pair-name>

Name of the key pair.

--key-file | -k <key-file-name>

Name of the key file that will store the private key of the key pair. If a path is not specified, then the command creates the file in the current directory.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates a key pair in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key-name	myKeyPair
key-file	myKeyFile

```
bash-3.2$ ./iaas-create-key-pair --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --key-name myKeyPair --key-file
myKeyFile
```

```
dx a9:60:cb:88:4a:42:2d:c5:d4:f1:23:63:64:54:d9:0a:e0:c5:a5:9e
bash-3.2$
```

Exit Status

(Success) – The command returns the key pair name and fingerprint. When using the [iaas-describe-key-pairs](#) command, the new key pair is listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-delete-key-pair](#)
- [iaas-describe-key-pairs](#)
- [iaas-import-key-pair](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-create-server-template-from-assembly

Creates a set of new server templates based on an assembly.

Syntax

```
iaas-create-server-template-from-assembly|iaas-cstfa [--base-url <base_url>]
                                                    [--debug] [--header] [--help]
                                                    [--access-key-file <access_key_file>]
                                                    --name <name> [--desc <descr>]
                                                    [--url] <url>
                                                    [--sep <separator>]
                                                    [--trust-store <truststore_file>]
                                                    [--verbose] [--xml]
```

Description

This command imports an assembly and creates one or more server templates in an account based on the assembly. Snapshots can be also created as a result of this action.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Add a header row to the output. If this option is not specified, then no header row is displayed.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--name | -n <name>

Name of the server template.

--desc | -d <descr>

Description of the server template.

--url | -u <url>

An assembly URL.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates a server template from the specified assembly. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myAssembly
url	http://myServer.com/myAssembly.ova

```
bash-3.2$ ./iaas-create-server-template-from-assembly --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file --name
myAssembly --url http://myServer.com/myAssembly.ova
```

```
ASSM-1110da89-64c7-4c11-a239-927b77e14c2
bash-3.2$
```

Exit Status

(Success) – The command returns the assembly ID and loads the server templates for an account. When the [iaas-describe-server-templates](#) command is used, the new server templates are listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-server-template-from-url](#)
- [iaas-create-server-template-from-vserver](#)
- [iaas-delete-server-template](#)
- [iaas-describe-attributes](#)
- [iaas-describe-server-templates](#)
- [iaas-describe-vservers](#)
- [iaas-modify-attributes](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-create-server-template-from-url

Creates a server template based on a file accessible through a URL.

Syntax

```
iaas-create-server-template-from-url|iaas-cstfu [--base-url <base_url>] [--debug]
                                         [--header] [--help]
                                         [--access-key-file <access_key_file>]
                                         --name <name> [--desc <descr>]
                                         [--url] <url>
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command downloads the server template from the specified URL and then loads the server template to the associated account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--name | -n <name>

Name of the server template.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--desc | -d <descr>

Description of the server template.

--url | -u <url>

Server template URL from which the server template is downloaded.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates a server template from the specified URL. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myST
url	http://myServer.com/vm-templates/OVM.tar.gz

```
bash-3.2$ ./iaas-create-server-template-from-url --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file --name myST
--url http://myServer.com/vm-templates/OVM.tar.gz
```

```
TMPL-9330da89-64c7-4c11-a239-927b77e344d1
bash-3.2$
```

Exit Status

(Success) – The command returns the server template ID and loads the server template in an account. When using the [iaas-describe-server-templates](#) command, the server template is listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-server-template-from-assembly](#)
- [iaas-create-server-template-from-vserver](#)
- [iaas-delete-server-template](#)
- [iaas-describe-attributes](#)
- [iaas-describe-server-templates](#)
- [iaas-describe-vservers](#)
- [iaas-modify-attributes](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-create-server-template-from-vserver

Creates a new server template based on a stopped vServer.

Syntax

```
iaas-create-server-template-from-vserver|iaas-cstfi [--base-url <base_url>]
                                                [--debug]
                                                [--header] [--help]
                                                [--access-key-file <access_key_file>]
                                                [--desc <descr>]
                                                --name <name>
                                                --vserver-id <vserver-id>
                                                [--sep <separator>]
                                                [--trust-store <truststore_file>]
                                                [--verbose] [--xml]
```

Description

This command creates a new server template based on a stopped vServer in the account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--desc | -d <descr>

Description of the template.

--name | -n <name>

Name of the template.

--vserver-id | -i <vserver-id>

vServer ID.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates a server template from the specified URL. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myST
vServer ID	VSRV-a959be32-6237-40c7-b44b-47d392e7a0ac

```
bash-3.2$ ./iaas-create-server-template-from-vserver --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file --name myST
--vserver-id VSRV-a959be32-6237-40c7-b44b-47d392e7a0ac
```

```
TMPL-00005a39-93fd-44d9-86e5-2755d506f16f
bash-3.2$
```

Exit Status

(Success) – The command returns the server template ID and loads the server template in the account. When using the [iaas-describe-server-templates](#) command, the server template is listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-server-template-from-assembly](#)
- [iaas-create-server-template-from-url](#)
- [iaas-delete-server-template](#)
- [iaas-describe-attributes](#)
- [iaas-describe-server-templates](#)
- [iaas-describe-vservers](#)
- [iaas-modify-attributes](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-create-snapshot

Creates a snapshot based on an existing volume for an account.

Syntax

```
iaas-create-snapshot|iaas-cs [--base-url <base_url>] [--debug]
                             [--access-key-file <access_key_file>]
                             [--header] [--help]
                             --name <name> [--desc <descr>]
                             [--volume-id] <volume-id>
                             [--sep <separator>]
                             [--trust-store <truststore_file>]
                             [--verbose] [--xml]
```

Description

This command creates a snapshot based on an existing volume.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--desc | -d <descr>

Description of the snapshot.

--name | -n <name>

Name of the snapshot.

--volume-id <volume-id>

ID of the volume.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example creates a snapshot from the specified volume ID. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
volume-id	VOL-246b5c62-4072-41cf-885b-99d6c63583bd

```
bash-3.2$ ./iaas-create-snapshot --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --volume-id
VOL-246b5c62-4072-41cf-885b-99d6c63583bd --name mySnapshot
```

```
SNAP-d7ca2baf-d10b-4ae9-a2c5-5355da9d7004
bash-3.2$
```

Example 2

This example creates a snapshot from the specified volume ID. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	ak.file
volume-id	NA	VOL-246b5c62-4072-41cf-885b-99d6c63583bd

```
bash-3.2$ ./iaas-create-snapshot --volume-id
VOL-246b5c62-4072-41cf-885b-99d6c63583bd --name mySnapshot2
```

```
SNAP-d7ca2baf-d10b-4ae9-a2c5-5355da9d7002
bash-3.2$
```

Exit Status

(Success) – The command returns the snapshot ID of the snapshot created. When using the [iaas-describe-snapshots](#) command, the snapshot is listed.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-volume](#)
- [iaas-delete-snapshot](#)

- [iaas-describe-snapshots](#)
- [iaas-describe-volumes](#)

iaas-create-tags

Creates tags for a resource in an account.

Syntax

```
iaas-create-tags|iaas-ctag [--base-url <base_url>] [--debug]
                          [--header] [--help]
                          [--access-key-file <access_key_file>]
                          --id <resource-id>
                          [--tags <name>=<val>[,<name>=<val>]*
                          [--sep <separator>]
                          [--trust-store <truststore_file>]
                          [--verbose] [--xml]
```

Description

This command adds or overwrites tags for the specified resource ID that exists in an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--id <resource-id>

ID of a resource in the account.

--tags <name>=<val>[,<name>=<val>]*

List of tag names and values.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example adds a tag to the specified virtual network ID. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
id	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t

```
bash-3.2$ ./iaas-create-tags --base-url https://<EnterpriseControllerHostname>/ -a
ak.file --id VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t --tags myTag=myTagValue
```

```
bash-3.2$
```

Example 2

This example adds two tags to the specified virtual network ID. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vNet ID	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t

```
bash-3.2$ ./iaas-create-tags --base-url https://<EnterpriseControllerHostname>/ -a
ak.file --id VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t --tags myTag=myTagValue
myTag2=myTag2Values
```

```
bash-3.2$
```

Exit Status

(Success) – The command adds or overwrites tags to the specified resource. This command does not return any value, only the command prompt is returned. You can see the attributes of the new tags using the [iaas-describe-tags](#) command.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-delete-tags](#)
- [iaas-describe-tags](#)

iaas-create-vnet

Creates a private virtual network for the account.

Syntax

```
iaas-create-vnet|iaas-cvn [--base-url <base_url>] [--debug]
                          [--header] [--help]
                          [--access-key-file <access_key_file>]
                          --name <name> [--desc <descr>]
                          [--size <size>]
                          [--sep <separator>]
                          [--trust-store <truststore_file>]
                          [--verbose] [--xml]
```

Description

This command creates a private virtual network for an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--name | -n <name>

Name of the virtual network.

--size | -s <size>

Size of the subnet of the virtual network.

--desc | -d <descr>

Description of the virtual network.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example creates a virtual network with the default subnet size. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myVNET

```
bash-3.2$ ./iaas-create-vnet --base-url https://<EnterpriseControllerHostname>/
--access-key-file ak.file --name myVNET
```

```
VNET-350c9c3f-0ee5-41be-917e-ebbaed0fa4ad
bash-3.2$
```

Example 2

This example creates a virtual network with a subnet size of 16. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	ak.file
name	NA	myVNET2
size	NA	16

```
bash-3.2$ ./iaas-create-vnet --name myVNET2 --size 16
```

```
VNET-480c9c3f-0ee5-41be-917e-ebbaed0fa4gt
bash-3.2$
```

Exit Status

(Success) – The command returns the vNet ID of the vNet created. You can see the attributes of the new vNet using the [iaas-describe-vnets](#) command.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-delete-vnet](#)
- [iaas-describe-ip-addresses](#)
- [iaas-describe-vnets](#)

- [iaas-release-ip-addresses](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-create-volume

Creates a volume for an account.

Syntax

```
iaas-create-volume|iaas-cvol [--base-url <base_url>] [--debug] [--header]
                             [--access-key-file <access_key_file>] [--help]
                             [--desc <descr>] [--snapshot-id <snapshot-id>]
                             --name <name> [--shared] [--size <size>]
                             [--sep <separator>]
                             [--trust-store <truststore_file>]
                             [--verbose] [--xml]
```

Description

This command creates an empty volume of the specified size for the account. This command also provides an option to create a volume from a snapshot.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--name | -n <name>

Name of the volume.

--shared

Specifies if the volume is a shared volume. The default is a nonshared volume.

--size | -s <size>

Size of the volume in gigabytes (GB). This option is required if a snapshot ID is not given.

--snapshot-id | -i <snapshot-id>

Snapshot ID. The volume is created based on the specified snapshot.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--desc | -d <descr>

Description of the volume.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates an empty volume of 2 GB. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	firstVol

```
bash-3.2$ ./iaas-create-volume --base-url https://<EnterpriseControllerHostname>/
--access-key-file ak.file --name firstVol --size 2
```

```
VOL-f23a8ba1-ec55-4159-bbdf-de220d18a1c4
bash-3.2$
```

Example 2

This example creates a volume based on the specified snapshot. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	ak.file
snapshot-id	NA	SNAP-7a717e39-fe67-4573-a93d-889b3446176b

```
bash-3.2$ ./iaas-create-volume --base-url https://<EnterpriseControllerHostname>/
--access-key-file ak.file --name myVol --snapshot-id
SNAP-7a717e39-fe67-4573-a93d-889b3446176b
```

```
VOL-g23a8ba1-ec55-4159-bbdf-de220d18a1b7
bash-3.2$
```

Exit Status

(Success) – The command returns the volume ID of the volume created. Attributes of the new volume are listed using the [iaas-describe-volumes](#) command.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-snapshot](#)
- [iaas-delete-snapshot](#)
- [iaas-delete-volume](#)
- [iaas-describe-snapshots](#)
- [iaas-describe-volumes](#)
- [iaas-import-volume](#)

iaas-delete-distribution-group

Deletes an existing distribution group.

Syntax

```
iaas-delete-distribution-group|iaas-deldg [--base-url <base_url>] [--debug]
                                         [--header] [--help]
                                         [--access-key-file <access_key_file>]
                                         [--distribution-group-id] <distribution group-id>
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command deletes a distribution group from an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

[--distribution-group-id] <distribution group-id>

ID of the distribution group.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example deletes an existing distribution group from an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
distribution-group-id	DG-3206ed4d-44c9-42c4-8df7-511709a49ead

```
bash-3.2$ ./iaas-delete-distribution-group --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file
--distribution-group-id DG-3206ed4d-44c9-42c4-8df7-511709a49ead
```

```
bash-3.2$
```

Exit Status

(Success) – The command deletes the specified distribution group from the account. The command does not return any value, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-distribution-group](#)
- [iaas-describe-distribution-groups](#)
- [iaas-run-vsriver](#)
- [iaas-run-vservers](#)

iaas-delete-key-pair

Deletes an existing key pair from an account.

Syntax

```
iaas-delete-key-pair|iaas-delkp  [--base-url <base_url>] [--debug]
                                  [--header] [--help]
                                  [--access-key-file <access_key_file>]
                                  --key-name <key-pair-name>
                                  [--sep <separator>]
                                  [--trust-store <truststore_file>]
                                  [--verbose] [--xml]
```

Description

This command deletes a key pair from an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--key-name <key-pair-name>

ID of the key pair.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example deletes an existing key pair from an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key-name	key1

```
bash-3.2$ ./iaas-delete-key-pair --base-url  
https://<EnterpriseControllerHostname>/ --access-key-file ak.file --key-name key1
```

```
bash-3.2$
```

Exit Status

(Success) – The command deletes the key pair. The command does not return any value, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-key-pair](#)
- [iaas-describe-key-pairs](#)
- [iaas-import-volume](#)

iaas-delete-server-template

Deletes an existing server template.

Syntax

```
iaas-delete-server-template|iaas-delstem [--base-url <base_url>] [--debug]
                                         [--header] [--help]
                                         [--access-key-file <access_key_file>]
                                         <server-template-id>
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command deletes the server template with the specified server template ID.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

[--server-template-id] <server-template-id>

ID of the server template.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example deletes an existing server template from an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
server-template-id	TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee03

```
bash-3.2$ ./iaas-delete-server-template --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file
--server-template-id TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee03
```

```
bash-3.2$
```

Exit Status

(Success) – The command deletes the server template. The command does not return any value, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-server-template-from-assembly](#)
- [iaas-create-server-template-from-url](#)
- [iaas-create-server-template-from-vserver](#)
- [iaas-describe-attributes](#)
- [iaas-describe-server-templates](#)

iaas-delete-snapshot

Deletes a snapshot from an account.

Syntax

```
iaas-delete-snapshot|iaas-dels [--base-url <base_url>] [--debug]
                                [--access-key-file <access_key_file>]
                                [--header] [--help]
                                [--snapshot-id <snapshot-id>]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command deletes a snapshot from an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--snapshot-id <snapshot-id>

ID of the snapshot

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example deletes the specified snapshot from an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
snapshot-id	SNAP-d743e90c-53c5-4b01-a297-e2c944755c48
access-key-file	ak.file

```
bash-3.2$ ./iaas-delete-snapshot --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --snapshot-id
SNAP-d743e90c-53c5-4b01-a297-e2c944755c48
```

```
bash-3.2$
```

Exit Status

(Success) – The command deletes the specified snapshot. The command does not return any value, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-snapshot](#)
- [iaas-create-volume](#)
- [iaas-delete-volume](#)
- [iaas-describe-snapshots](#)
- [iaas-describe-volumes](#)

iaas-delete-tags

Deletes tags from a resource.

Syntax

```
iaas-delete-tags|iaas-deltag [--base-url <base_url>] [--debug]
                             [--header] [--help]
                             [--access-key-file <access_key_file>]
                             --id <resource-id>
                             [--tags] <name|val>[,<name|val>]*
                             [--sep <separator>]
                             [--trust-store <truststore_file>]
                             [--verbose] [--xml]
```

Description

This command deletes tags from a resource in an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--id <resource-id>

ID of a resource in an account.

--tags <name>=<val>[,<name>=<val>]*

List of tag names or values of the tags to be deleted.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example deletes a tag from the specified resource. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
id	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t
tags	myTag

```
bash-3.2$ ./iaas-delete-tags --base-url https://<EnterpriseControllerHostname>/ -a
ak.file --id VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t --tags myTag
```

```
bash-3.2$
```

Example 2

This example deletes two tags from the specified resource. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
id	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t
tags	myTag,myTag2

```
bash-3.2$ ./iaas-delete-tags --base-url https://<EnterpriseControllerHostname>/ -a
ak.file --id VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t --tags myTag,myTag2
```

```
bash-3.2$
```

Exit Status

(Success) – The command deletes tags from the specified resource. This command does not return any value as part of the output, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-tags](#)
- [iaas-describe-tags](#)

iaas-delete-vnet

Deletes a private vNet.

Syntax

```
iaas-delete-vnet|iaas-delvn [--base-url <base_url>] [--debug]
                             [--header] [--help]
                             [--access-key-file <access_key_file>]
                             [--vnet] <vnet-id>
                             [--sep <separator>]
                             [--trust-store <truststore_file>]
                             [--verbose] [--xml]
```

Description

This command deletes a private vNet from an account. The vNet must not be associated with a vServer.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vnet <vnet-id>

ID of the virtual network.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example deletes the specified virtual network. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vnet	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t

```
bash-3.2$ ./iaas-delete-vnet --base-url https://<EnterpriseControllerHostname>/ -a  
ak.file --vnet VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t
```

```
bash-3.2$
```

Exit Status

(Success) – The command deletes the specified vNet. This command does not return any response, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-vnet](#)
- [iaas-describe-ip-addresses](#)
- [iaas-describe-vnets](#)
- [iaas-release-ip-addresses](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-delete-volume

Deletes a volume from an account.

Syntax

```
iaas-delete-volume|iaas-delvol [--base-url <base_url>] [--debug] [--header]
                                [--access-key-file <access_key_file>] [--help]
                                [--volume-id <volume-id>]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command deletes a volume. A deletion of a volume does not affect any volume snapshots that have been created prior based on the deleted volume.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--volume-id <volume-id>

ID of the volume.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example deletes the specified volume. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
volume ID	VOL-82b4661e-de12-4931-b9a4-cce8edd23532

```
bash-3.2$ ./iaas-delete-volume --base-url https://<EnterpriseControllerHostname>/  
-a ak.file --volume-id VOL-82b4661e-de12-4931-b9a4-cce8edd23532
```

```
bash-3.2$
```

Exit Status

(Success) The command deletes the specified volume. This command does not return any response, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-snapshot](#)
- [iaas-create-volume](#)
- [iaas-delete-snapshot](#)
- [iaas-describe-snapshots](#)
- [iaas-describe-volumes](#)

iaas-describe-attributes

Displays attributes of a resource in an account.

Syntax

```
iaas-describe-attributes|iaas-dat [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                --id <resource-id>
                                [--attr-names] <name>[,<name>]*
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command displays the attributes of a resource in an account. This command requires the input of the ID of the resource and the name of the attribute.

The name of the attribute is displayed as part of the header in the describe command for the related resource. For example, you can use the [iaas-describe-volumes](#) command with the `--header` option to see the attribute names for volumes.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--id <resource-id>

The ID of the resource. The ID of the resource can be obtained using the corresponding `iaas-describe*` command, for example `iaas-describe-server-template`.

--attr-names <name>[,<name>]*

List of attribute names. The name of the attribute is any of the names displayed in the header of the output of an `iaas-describe-*` command (for example, `iaas-describe-server-template -H`).

--sep <separator>

Column separator character. The default separator is *TAB*.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example displays the name attribute of the specified vNet. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
id	VNET-7403e87f-1bab-4097-98ae-ea72d8fe4b3f

```
bash-3.2$ ./iaas-describe-attributes --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --id
VNET-7403e87f-1bab-4097-98ae-ea72d8fe4b3f --attr-names name

name myVnet2
bash-3.2$
```

Example 2

This example displays the name and status attributes of the specified server template. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
id	TMPL-aaaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01

```
bash-3.2$ ./iaas-describe-attributes --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --id
TMPL-aaaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01 --attr-names name,status

name    default-server-templatel
status  OK
bash-3.2$
```

Exit Status

(Success) – The command returns the list of attributes of the specified resource.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-server-templates](#)
- [iaas-describe-snapshots](#)
- [iaas-describe-vnets](#)
- [iaas-describe-volumes](#)
- [iaas-describe-vservers](#)
- [iaas-modify-attributes](#)

iaas-describe-distribution-groups

Lists distribution groups for an account.

Syntax

```
iaas-describe-distribution-groups | iaas-ddg [--base-url <base_url>] [--debug]
                                         [--access-key-file <access_key_file>]
                                         [--help] [--header]
                                         [--filters <name>=<val>[,<name>=<val>]*]
                                         [--ids <dg-id>[,<dg-id>]*]
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command lists the distribution groups in an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters | -f <name>=<val>[,<name>=<val>]*

List of filters.

--ids | -i <dg-id>[,<dg-id>]*

List of distribution group IDs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example lists all the distribution groups in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-distribution-groups --base-url
https://<EnterpriseControllerHostname>/ -a ~/ak.file -H
```

```
id      name      description      status      vservers      size
DG-8e7b0d64-57ca-492d-806d-a2db6b9a00df myDistributionGroup2      OK
VSRV-0f9b005f-bd7d-4166-a4db-5d81cabf5c89      10
DG-068ae84c-d0fc-406d-aa37-0be4f88d411c myDistributionGroup      OK
50000
bash-3.2$
```

Exit Status

(Success) – The command returns the list of distribution groups and their attributes. If no distribution groups are found, then the response is empty.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-distribution-group](#)
- [iaas-delete-distribution-group](#)

iaas-describe-ip-addresses

Lists of the IP addresses allocated in an account.

Syntax

```
iaas-describe-ip-addresses|iaas-dip  [--base-url <base_url>] [--debug]
                                     [--access-key-file <access_key_file>]
                                     [--help] [--header]
                                     [--filters <name>=<val>[,<name>=<val>]*]
                                     [--sep <separator>]
                                     [--trust-store <truststore_file>]
                                     [--verbose] [--xml]
```

Description

This command lists the IP addresses allocated in an account. The attributes displayed as part of the list include the IP address, the vNet ID, and the vServer ID if the IP address is associated with a vServer.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <name>=<val>[,<name>=<val>]*

List of filters. Valid names are *ipAddress*, *vnet*, and *vserver*.

--sep <separator>

Column separator character. The default separator is *TAB*.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example lists all allocated IP addresses for an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-ip-addresses --base-url
https://<EnterpriseControllerHostname>/ -a ak.file
```

```
192.168.2.5 VNET-cde375904-9d34-4ec8-86b9-d917712abaaa
VSRV-123456789-1234-1234-1234-123456789123
192.168.2.1 VNET-vfr4375904-9d34-4ec8-86b9-d917712abaaa
bash-3.2$
```

Example 2

This example lists the allocated IP addresses from the specified virtual network. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vnet	VNET-cde375904-9d34-4ec8-86b9-d917712abaaa

```
bash-3.2$ ./iaas-describe-ip-addresses --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --filters
vnet=VNET-cbd75904-9d34-4ec8-86b9-d917712abaaa
```

```
192.168.2.5 VNET-cde375904-9d34-4ec8-86b9-d917712abaaa
VSRV-123456789-1234-1234-1234-123456789123
192.168.2.1 VNET-cde375904-9d34-4ec8-86b9-d917712abaaa
bash-3.2$
```

Exit Status

(Success) – The command returns the list of allocated IP addresses and their attributes. If no IP addresses are found, then the response is empty.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-allocate-ip-addresses](#)
- [iaas-create-vnet](#)
- [iaas-describe-vnets](#)
- [iaas-release-ip-addresses](#)

iaas-describe-key-pairs

Displays the available key pairs for an account.

Syntax

```
iaas-describe-key-pairs|iaas-dkp [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                [--filters <name>=<val>[,<name>=<val>]*]
                                [--key-names <key-pair-names>]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command lists the key pairs for an account. This command also allows filtering of the key pairs displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <name>=<val>[,<name>=<val>]*

List of filters.

--key-name|-n <key-pair-name>

Name of key pairs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example lists the attributes of all the existing key pairs in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-key-pairs --base-url
https://<EnterpriseControllerHostname>/ -a ak.file
```

```
key1    d8:ab:f4:ca:51:ee:40:61:a1:c2:c0:13:49:7c:23:77
bash-3.2$
```

Example 2

This example lists the attributes of the specified key pair. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key pair name	key1

```
bash-3.2$ ./iaas-describe-key-pairs --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --key-names key1 -H
```

```
name    fingerprint
key1    d8:ab:f4:ca:51:ee:40:61:a1:c2:c0:13:49:7c:23:77
bash-3.2$
```

Exit Status

(Success) – The command returns the list of existing key pairs and their attributes. If no key pairs are found, then the response is empty.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-key-pair](#)
- [iaas-delete-key-pair](#)
- [iaas-import-key-pair](#)
- [iaas-run-vsserver](#)

- [iaas-run-vservers](#)

iaas-describe-server-templates

Lists server templates and their attributes for an account.

Syntax

```
iaas-describe-server-templates|iaas-dstem [--base-url <base_url>] [--debug]
                                         [--access-key-file <access_key_file>]
                                         [--header] [--help]
                                         [--filters <name>=<val>[,<name>=<val>]*]
                                         [--ids <t-id>[,<t-id>]*]
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command lists the server templates and their attributes associated with the account. This command also allows filtering of the server templates displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <name>=<val>[,<name>=<val>]*

List of filters to limit the server templates listed by this command. Valid names for filters are description, ID, imageType, memory, name, public, readOnly, size, and status.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--ids|-i <t-id>[,<t-id>]*

List of server template IDs to filter the server templates listed by this command.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example displays all existing server templates of an account and adds a header row. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-server-templates --base-url
https://<EnterpriseControllerHostname>/ --access-key-file ak.file -H

id      name      description      status size      public imageType      readOnly
TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee03 default-server-template3 Default Server
Template OK 197912092999680 false VDMK false
TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee05 default-server-template5 Default Server
Template OK 197912092999680 false VDMK false
bash-3.2$
```

Example 2

This example displays server templates that are public to other accounts. It uses the following options and values:

Option	Environment Variable	Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	<username>
filters	NA	public=true

```
bash-3.2$ ./iaas-describe-server-templates --filters public=true

TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee02      default-server-template2
Default Server Template OK      197912092999680 true  VDMK  false
TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01      default-server-template1
Default Server Template OK      197912092999680 true  VDMK  false
bash-3.2$
```

Exit Status

(Success) – The command returns a list of the found server templates.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-server-template-from-assembly](#)
- [iaas-create-server-template-from-url](#)

- [iaas-create-server-template-from-vserver](#)
- [iaas-delete-server-template](#)
- [iaas-describe-attributes](#)
- [iaas-describe-vservers](#)
- [iaas-modify-attributes](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-describe-snapshots

Displays the snapshots for an account.

Syntax

```
iaas-describe-snapshots|iaas-ds [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                [--filters <val>[,<name>=<val>]*]
                                [--ids <vs-id>[,<vs-id>]*]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command lists the available snapshots for an account. The command also allows filtering of the snapshots displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <val>[,<name>=<val>]*

List of filters. Valid names are description, id, name, and status.

--ids|-i <vs-id>[,<vs-id>]*

IDs of the snapshots.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example lists the attributes of all the existing snapshots in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-snapshots --base-url
https://<EnterpriseControllerHostname>/ -a ak.file -H
```

```
id      name      description      status
SNAP-2f2039cd-943b-4072-9ded-e96b54b7ca79      mySnapshot2      OK
SNAP-d743e90c-53c5-4b01-a297-e2c944755c48      mySnapshot      OK
bash-3.2$
```

Exit Status

(Success) – The command returns a list of the snapshots and their attributes.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-snapshot](#)
- [iaas-create-volume](#)
- [iaas-delete-snapshot](#)
- [iaas-describe-volumes](#)

iaas-describe-tags

Displays the existing tags in an account.

Syntax

```
iaas-describe-tags|iaas-dtag [--base-url <base_url>] [--debug]
                             [--header] [--help]
                             [--access-key-file <access_key_file>]
                             [--filters <name>[=<val>][,<name>[=<val>]]*]
                             [--ids <res-id>[,<res-id>]*]
                             [--sep <separator>]
                             [--trust-store <truststore_file>]
                             [--verbose] [--xml]
```

Description

This command lists the resources in an account that have a tag associated with them. The attributes displayed by this command are the resource ID, the name of the tag, and the value of the tag. The command also allows filtering of the tags displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <name>[=<val>][,<name>[=<val>]]*

List of filters to limit the resources listed by this command.

Allows the filtering of tags by tag name, tag value, and resource identifier. The result contains only tags that match all elements of the filter list.

--ids|-i <res-id>[,<res-id>]*

IDs of resources in the account to limit the output to only those tags associated with these resources.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example lists the resources that have a tag associated with them in an account, along with the tag name and value. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-tags --base-url https://<EnterpriseControllerHostname>/
-a ak.file
```

```
VNET-77d2b71b-a5ff-4d76-bdc4-fbd9317bbdc1      vNet1Tag  vNet1TagValue
VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t     vNet2Tag  vNet1TagValue
bash-3.2$
```

Example 2

This example lists the tag names and values of the specified virtual network. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
ids	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t

```
bash-3.2$ ./iaas-describe-tags --base-url https://<EnterpriseControllerHostname>/
-a ak.file --ids VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t
```

```
VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t      vNet2Tag  vNet1TagValue
bash-3.2$
```

Example 3

This example identifies server templates from a particular assembly filtering with the tag name assemblyKey. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

Option	Option Value
filters	assemblyKey=ASSM-ad6c7baa-0373-41bd-bd5a-051e4ec5f896

```
bash-3.2$ ./iaas-describe-tags --base-url https://<EnterpriseControllerHostname>/
-a ~/ak.file -H --ids
TMPL-955e701d-1396-4e9a-b427-45b278870e1b, TMPL-fa591348-bfd1-40a3-b3a2-e030af76b96
8, TMPL-bc78b1c1-3a78-4f78-a486-6b2849e8040a, TMPL-5179af29-4332-434f-a25f-8002e61e8
7ee --filters assemblyKey=ASSM-ad6c7baa-0373-41bd-bd5a-051e4ec5f896

resourceId      name      value
TMPL-955e701d-1396-4e9a-b427-45b278870e1b      assemblyKey
ASSM-ad6c7baa-0373-41bd-bd5a-051e4ec5f896
TMPL-fa591348-bfd1-40a3-b3a2-e030af76b968      assemblyKey
ASSM-ad6c7baa-0373-41bd-bd5a-051e4ec5f896
bash-3.2$
```

Exit Status

(Success) – The command returns the list of resources with the name and value of the associated tag.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-tags](#)
- [iaas-delete-tags](#)

iaas-describe-vdc-capabilities

Displays the vDC capabilities for an account.

Syntax

```
iaas-describe-vdc-capabilities|iaas-dvcap [--base-url <base_url>] [--debug]
                                     [--header] [--help]
                                     [--access-key-file <access_key_file>]
                                     [--filters <name>[=<val>][,<name>[=<val>]]*]
                                     [--sep <separator>]
                                     [--trust-store <truststore_file>]
                                     [--verbose] [--xml]
```

Description

This command displays the virtual datacenter (vDC) capabilities for an account. It also allows filtering of the vDC capabilities displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--filters|-f <name>[=<val>][,<name>[=<val>]]*

List of capability names and optional values to filter.

--help | -h

Displays the usage information for this command.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example lists the capabilities of the vDC associated with the account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-vdc-capabilities --base-url
https://<EnterpriseControllerHostname>/ -a ak.file -H

name    value
VirtualizationType    OVM
VirtualizationVersion  3.0.2
ProcessorArch    Default_Intel_F6_M23
ProcessorVersion    Default_Intel_Family:6_Model:23
DistributionGroupSupport    disabled
HighAvailabilityUserControl=disabled
HighAvailabilityDefault=true
bash-3.2$
```

Example 2

This example lists only the virtualization type of the vDC for an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
filters	VirtualizationType

```
bash-3.2$ ./iaas-describe-vdc-capabilities --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --filters VirtualizationType

VirtualizationType    OVM
bash-3.2$
```

Exit Status

(Success) – The command returns the list of vDC capabilities of the account.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-delete-distribution-group](#)
- [iaas-run-vsriver](#)
- [iaas-run-vsriver](#)

iaas-describe-vnets

Lists virtual network information for an account.

Syntax

```
iaas-describe-vnets | iaas-dvn  [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                [--filters <name>=<val>[,<name>=<val>]*]
                                [--ids <vn-id>[,<vn-id>]*]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command lists the available virtual networks for an account. It also allows filtering of the vNets displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters | -f <name>=<val>[,<name>=<val>]*

List of filters to limit the virtual networks listed by this command. Valid names for filters are `id`, `name`, `description`, `status`, and `ipAddress`. The result contains only virtual networks that match all elements of the filter list.

--ids <vn-id>[,<vn-id>]*

IDs of the virtual networks.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example lists the attributes of all the virtual networks in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-vnets --base-url https://<EnterpriseControllerHostname>/
-a ak.file
```

```
VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c      vnet1 descvnet1 OK
192.168.0.0/24
VNET-8521f019-b50d-4051-8028-7ed2b0f5d767    vnet2 descvnet2 OK
192.168.3.0/24
VNET-7418613f-efaa-42f4-bc96-9583ec39a481    vnet3 descvnet3 OK
192.168.5.0/24
VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t    vnet4 descvnet4 OK
192.168.6.0/24
bash-3.2$
```

Example 2

This example lists the attributes of the specified virtual network IDs. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
ids	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t VNET-7418613f-efaa-42f4-bc96-9583ec39a481

```
bash-3.2$ ./iaas-describe-vnets --base-url https://<EnterpriseControllerHostname>/
-a ak.file --ids
VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t,VNET-7418613f-efaa-42f4-bc96-9583ec39a48
1
```

```
VNET-7418613f-efaa-42f4-bc96-9583ec39a481      vnet3 descvnet3 OK
192.168.5.0/24
VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t    vnet4 descvnet4 OK
192.168.6.0/24
bash-3.2$
```

Exit Status

(Success) – The command returns the list of existing virtual networks and their attributes. If no virtual networks are found, the response is empty.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-vnet](#)
- [iaas-delete-vnet](#)
- [iaas-describe-ip-addresses](#)
- [iaas-release-ip-addresses](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-describe-volumes

Displays volume information for an account.

Syntax

```
iaas-describe-volumes|iaas-dvol [--base-url <base_url>] [--debug]
                                [--access-key-file <access_key_file>]
                                [--filters <name>=<val>[,<name>=<val>]*]
                                [--ids] <vol-id>[,<vol-id>]*
                                [--sep <separator>] [--header] [--help]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command lists the available volumes for an account. This command also allows filtering of the volumes displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <name>=<val>[,<name>=<val>]*

List of filters. Valid names are: `id`, `status`, `name`, `description`, `size`, `shared`, and `vservers`.

--ids|-i <vol-id>[,<vol-id>]*

List of volume IDs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example lists the attributes of all the volumes found in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-volumes --base-url
https://<EnterpriseControllerHostname>/ -a ak.file -H
id      name      description      status  size      shared  vservers
VOL-9dc9039c-012c-4b4b-9ea8-533ec0678eb9      testvolumeshared      this is a
shared volume OK  4.0      true
VSRV-5100c512-fc39-4a59-ade9-653a1e041c10,VSRV-76ed9c11-d5d5-4418-833d-b2025c7bfdeb
VOL-3469f96d-d1bf-4a22-8b6c-5046df2d66d7      test-volume      OK
16.0    false    VSRV-76ed9c11-d5d5-4418-833d-b2025c7bfdeb
VOL-87d9bd57-0ac1-443c-8e44-67daeb42412b      testvolumeunused      OK
4.0     false
```

Example 2

This example lists the attributes of the specified volume IDs. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
ids	VOL-7f7b1963-283c-4f2d-9748-30cf87119cef VOL-82b4661e-de12-4931-b9a4-cce8edd23532

```
bash-3.2$ ./iaas-describe-volumes --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --ids
VOL-7f7b1963-283c-4f2d-9748-30cf87119cef,VOL-82b4661e-de12-4931-b9a4-cce8edd23532

VOL-7f7b1963-283c-4f2d-9748-30cf87119cef myVol1 DescmyVol1 OK 4.0      false
VOL-82b4661e-de12-4931-b9a4-cce8edd23532 myVol2 DescmyVol1 OK 8.0      false
VSRV-76ed9c11-d5d5-4418-833d-b2025c7bfdeb
bash-3.2$
```

Exit Status

(Success) – The command returns the list of existing volumes and their attributes. If no volumes are found, the response is empty.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-snapshot](#)
- [iaas-create-volume](#)
- [iaas-delete-snapshot](#)
- [iaas-delete-volume](#)
- [iaas-describe-snapshots](#)

iaas-describe-vserver-metrics

Displays vServers metrics for an account.

Syntax

```
iaas-describe-vserver-metrics|iaas-dvm [--access-key-file <access_key_file>]
                                         [--base-url <base_url>] [--debug]
                                         [--header] [--help]
                                         [--filters <name>=<val>[,<name>=<val>]*]
                                         [--ids <id>[,<id>]*]
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command displays attributes of vServers in an account. The command also allows filtering of vServers displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <name>=<val>[,<name>=<val>]*

List of filters. Valid filter names are: `description`, `name`, `id`, and `cpuUsage`.

--ids|-i <id>[,<id>]*

List of vServer IDs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example lists the attributes of all the vServers found in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-vserver-metrics --base-url
https://<EnterpriseControllerHostname>/ -a ak.file -H
```

```
id      name      description      status  cpuUsage
VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64  testvserver3  Oracle VM Virtual
Machine    RUNNING    0.1508
VSRV-71f63b37-d61e-4900-a1a5-2b7d51b029c9  testvserver5  Oracle VM Virtual
Machine    RUNNING    0.067
bash-3.2$
```

Example 2

This example lists the attributes of the specified vServers filtering by vServer name. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
filters	name=testvserver3

```
bash-3.2$ ./iaas-describe-vserver-metrics --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --filters name="testvserver3"
-H
```

```
id      name      description      status  cpuUsage
VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64  testvserver3  Oracle VM Virtual
Machine    RUNNING    0.1508
bash-3.2$
```

Exit Status

(Success) – The command returns the list of existing vServers and their metrics. If no vServers are found, the response is empty.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-reboot-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-describe-vservers

Displays vServer information for an account.

Syntax

```
iaas-describe-vservers|iaas-dvss  [--base-url <base_url>] [--debug]
                                   [--header] [--help]
                                   [--access-key-file <access_key_file>]
                                   [--filters <name>=<val>[,<name>=<val>]*]
                                   [--ids <id>[,<id>]*]
                                   [--sep <separator>]
                                   [--trust-store <truststore_file>]
                                   [--verbose] [--xml]
```

Description

This command displays the attributes of vServers in an account. The command also allows filtering of vServers displayed.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--filters|-f <name>=<val>[,<name>=<val>]*

List of filters. Valid filter names are: `description`, `name`, `status`, `id`, `serverTemplateId`, `keyPair`, `vserverType`, `ha`, `distributionGroup`, `volumes`, `vcpu`, `memoryMb`, `dedicatedStorageMb`, and `attachedStorageMb`.

--ids|-i <id>[,<id>]*

List of vServer IDs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example lists the attributes of all the vServers found in an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-vservers --base-url
https://<EnterpriseControllerHostname>/ -a ak.file -H

id      name      description      status vnets  ipAddresses  serverTemplateId
keyName vserverType  ha      distributionGroup  volumes vcpu  memoryMb
dedicatedStorageMb  attachedStorageMb
VSRV-5100c512-fc39-4a59-ade9-653ale041c10      testvserver2  Oracle VM Virtual
Machine      RUNNING      VNET-9bdbf831-5404-48c0-99c7-ala8fc879ca3
10.162.45.55  TEMPL-c11f3a91-80c1-4f28-b54c-23331ddd398      597  true
VOL-9dc9039c-012c-4b4b-9ea8-533ec0678eb9      2      8192  6443499520
3221225472
VSRV-76ed9c11-d5d5-4418-833d-b2025c7bfdeb      testvserver  Oracle VM Virtual
Machine      RUNNING      VNET-9bdbf831-5404-48c0-99c7-ala8fc879ca3
10.162.45.54  TEMPL-c11f3a91-80c1-4f28-b54c-23331ddd398      597  true
VOL-3469f96d-d1bf-4a22-8b6c-5046df2d66d7,VOL-9dc9039c-012c-4b4b-9ea8-533ec0678eb9
1      4096  6443499520  7516192768
VSRV-0f9b005f-bd7d-4166-a4db-5d81cabf5c89      testvserver4  vserver for the
distribution group  RUNNING  VNET-9bdbf831-5404-48c0-99c7-ala8fc879ca3
10.162.45.56  TEMPL-c11f3a91-80c1-4f28-b54c-23331ddd398      597  true
DG-b534de55-8abe-4eb8-a5a3-918a84d39c8f      1      4096  6443499520
0
VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64      testvserver3  Oracle VM Virtual
Machine      RUNNING  VNET-025a35bf-828d-4e3f-a0f4-76f3d9a73c42      192.168.0.18
TEMPL-557952b6-0b00-4a00-a2ca-ada480d99cc6598  597  true
VOL-dc80ef69-109d-4e7d-968d-2fa90b68db1b      2      8192  6443499520
3221225472
VSRV-71f63b37-d61e-4900-a1a5-2b7d51b029c9      testvserver5  Oracle VM Virtual
Machine      RUNNING  VNET-025a35bf-828d-4e3f-a0f4-76f3d9a73c42      192.168.0.19
TEMPL-557952b6-0b00-4a00-a2ca-ada480d99cc6597  597  true      1
4096  6443499520  0
bash-3.2$
```

Example 2

This example lists the vServers that have a total of storage attached of 3221225472. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/

Option	Option Value
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-vservers --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --filters
attachedStorageMb=3221225472 -H
```

```
id      name      description      status vnets  ipAddresses      serverTemplateId
keyName vserverType  ha      distributionGroup  volumes vcpu      memoryMb
dedicatedStorageMb      attachedStorageMb
VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64      testvserver2  Oracle VM Virtual
Machine      RUNNING VNET-025a35bf-828d-4e3f-a0f4-76f3d9a73c42      192.168.0.18
TMPL-557952b6-0b00-4a00-a2ca-ada480d99cc6598      true
VOL-dc80ef69-109d-4e7d-968d-2fa90b68db1b      2      8192      6443499520
3221225472
VSRV-2e4f6688-2c57-4f41-8d3d-83d2d71d1a64      testvserver3  Oracle VM Virtual
Machine      RUNNING VNET-025a35bf-828d-4e3f-a0f4-76f3d9a73c42      192.168.0.18
TMPL-557952b6-0b00-4a00-a2ca-ada480d99cc6598      true
VOL-dc80ef69-109d-4e7d-968d-2fa90b68db1b      2      8192      6443499520
3221225472
bash-3.2$
```

Exit Status

(Success) – The command returns the list of existing vServers and their attributes. If no vServers are found, the response is empty.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vserver-metrics](#)
- [iaas-reboot-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-describe-vserver-types

Displays the available vServer types for an account.

Syntax

```
iaas-describe-vserver-types|iaas-dvst  [--base-url <base_url>] [--debug]
                                         [--header] [--help]
                                         [--access-key-file <access_key_file>]
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command displays the available vServers types for an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax: `https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example lists all the permitted vServer types for an account. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

```
bash-3.2$ ./iaas-describe-vserver-types --base-url
https://<EnterpriseControllerHostname>/ -a ak.file -H
```

```
id      name    description      status  memorySize      storageSize      vcpu
small   small   small instance type      1073741824      10737418240     1
medium  medium  medium instance type      4294967296      107374182400    2
large   large   large instance type      17179869184     1073741824000   4
bash-3.2$
```

Exit Status

(Success) – The command returns the list of existing vServer types and their attributes. If no vServer types are found, then response is empty. If the response is an empty list, contact your cloud administrator to verify your cloud user’s privileges.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-detach-volumes-from-vserver

Detaches one or more volumes from a vServer.

Syntax

```
iaas-detach-volumes-from-vserver | iaas-detvol [--base-url <base_url>] [--debug]
                                           [--access-key-file <access_key_file>]
                                           [--header] [--help]
                                           [--force]
                                           --vserver-id <vserver-id>
                                           [--volume-ids] <vol-id>[,<vol-id>]*
                                           [--sep <separator>]
                                           [--trust-store <truststore_file>]
                                           [--verbose] [--xml]
```

Description

This command detaches one or more volumes from a vServer.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vserver-id | -i <vserver-id>

ID of the vServer.

[--volume-ids] <vol-id>[,<vol-id>]*

Volume IDs.

--force|-f

Force the detach operation.

--sep <separator>

Column separator character. The default separator is *TAB*.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example detaches the specified volume from a vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vserver-id	VSRV-0fb57293-347c-4717-96ef-6dd23154596f

```
bash-3.2$ ./iaas-detach-volumes-from-vserver --base-url
https://<EnterpriseControllerHostname>/ -a ~/ak.file --vserver-id
VSRV-0fb57293-347c-4717-96ef-6dd23154596f --volume-ids
VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15 --force
```

```
bash-3.2$
```

Exit Status

(Success) – The command detaches the specified volumes from the vServer. This command does not return any response, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-attach-volumes-to-vserver](#)
- [iaas-describe-volumes](#)
- [iaas-describe-vservers](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)

iaas-import-key-pair

Imports an existing key pair to an account.

Syntax

```
iaas-import-key-pair|iaas-ikp [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                --key-file <key-file-name>
                                --key-name <key-pair-name>
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command stores a public RSA key to the application. This public key is created with a third-party tool, and it is stored in the *key-file* used as part of this command.

The supported formats of the public key to be imported are:

- OpenSSH
- SECSH

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--key-file|-k <key-file-name>

Name of the file containing the public key.

--key-name|-n <key-pair-name>

Name of the key pair.

--sep <separator>

Column separator character. The default separator is *TAB*.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example imports a key pair based on the specified key pair file. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key-file	myOtherPair.key
key-name	myOtherPair

```
bash-3.2$ ./iaas-import-key-pair --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --key-name myOtherPair
--key-file myOtherPair.key
```

```
myOtherPair    d9:ab:f4:ca:51:ee:40:61:a1:c2:c0:13:49:7c:23:77
bash-3.2$
```

Exit Status

(Success) – The command returns the name and the fingerprint of the imported key.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-key-pair](#)
- [iaas-delete-key-pair](#)
- [iaas-describe-key-pairs](#)
- [iaas-run-vsserver](#)
- [iaas-run-vservers](#)

iaas-import-volume

Imports a volume.

Syntax

```
iaas-import-volume|iaas-ivol  [--base-url <base_url>] [--debug]
                                [--access-key-file <access_key_file>]
                                [--header] [--help]
                                --name <name>
                                [--desc <descr>]
                                [--url] <url>
                                [--sep <separator>]
                                [--shared]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command imports a volume from another location.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--name | -n <name>

Name of the volume.

--url <>

URL of the volume to import.

--sep <separator>

Column separator character. The default separator is `TAB`.

--shared

Flag to indicate that the volume is shared.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is \$HOME/.oracle_iaas/truststore.

--desc | -d <descr>

Description of the volume.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example imports a volume based on the image file of the specified URL. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
url	http://ovm.oracle.com/volume-image/volume.img
volume name	myVolume

```
bash-3.2$ ./iaas-import-volume --base-url https://<EnterpriseControllerHostname>/
-a ak.file --name myVolume --url http://ovm.oracle.com/volume-image/volume.img
```

```
VOL-e9afec8c-dbe2-4e03-8561-15716650b81e
bash-3.2$
```

Exit Status

(Success) – The command returns the volume ID.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-create-snapshot](#)
- [iaas-create-volume](#)
- [iaas-delete-snapshot](#)
- [iaas-delete-volume](#)
- [iaas-describe-snapshots](#)
- [iaas-describe-volumes](#)

iaas-modify-attributes

Modifies the attributes of a resource in an account.

Syntax

```
iaas-modify-attributes|iaas-ma [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                --id <resource-id>
                                [--attributes] <name>=<val>[,<name>=<val>]*
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command modifies the attributes of a resource in an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--help | -h

Displays the usage information for this command.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--id <resource-id>

ID of the resource. The ID of the resource can be obtained by using the corresponding `iaas-describe*` command (for example, `iaas-describe-server-templates`).

--attributes <name>=<val>[,<name>=<val>]*

List of attribute names. The name of the attribute is any of the names displayed in the header of output of an `iaas-describe-*` command (for example, `iaas-describe-server-templates -H`).

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example modifies the name attribute of the specified vNet. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
id	VNET-7403e87f-1bab-4097-98ae-ea72d8fe4b3f

```
bash-3.2$ ./iaas-modify-attributes --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --id
VNET-7403e87f-1bab-4097-98ae-ea72d8fe4b3f --attributes name=myNewName
```

```
bash-3.2$
```

Example 2

This example modifies the name and public attribute of the specified server template. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
id	TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01

```
bash-3.2$ ./iaas-modify-attributes --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --id
TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01 --attributes name=myNewName,public=true
```

```
bash-3.2$
```

Exit Status

(Success) – The command modifies the specified attributes. This command does not return any value, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-server-templates](#)
- [iaas-describe-snapshots](#)
- [iaas-describe-vnets](#)
- [iaas-describe-volumes](#)

- [iaas-describe-vservers](#)

iaas-reboot-vservers

Reboots vServers for an account.

Syntax

```
iaas-reboot-vservers|iaas-rebvss [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                [--vserver-ids <id>[,<id>]*]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command reboots one or more vServers for an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--help | -h

Displays the usage information for this command.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--vserver-ids <id>[,<id>]*

vServer IDs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example reboots a vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vserver-ids	VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6

```
bash-3.2$ ./iaas-reboot-vservers --base-url https://<EnterpriseControllerHostname>/  
-a ak.file --vserver-ids VSRV-0fb57293-347c-4717-96ef-6dd23154596f
```

```
bash-3.2$
```

Exit Status

(Success) The command restarts the specified vServers. This command does not return any response, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-receive-message-from-vserver

Receives one message from a vServer.

Syntax

```
iaas-receive-message-from-vserver|iaas-rmfvs [--base-url <base_url>] [--debug]
      [--header] [--help]
      [--access-key-file <access_key_file>]
      --vserver-id <vserver-id>
      [--key] <key>
      [--sep <separator>]
      [--trust-store <truststore_file>]
      [--verbose] [--xml]
```

Description

This command receives a message from a vServer. The command reads the message from the common buffer in the virtualization layer by name. This action might return an error if the vServer is not in an appropriate state to receive messages.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vserver-id | -i <vserver-id>

vServer ID.

--key <key>

Key of the message to retrieve.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example receives a message from a vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vServer ID	VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6

```
bash-3.2$ ./iaas-receive-message-from-server --base-url
https://<EnterpriseControllerHostname>/ -a ~/ak.file --vserver-id
VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6 --key com.oracle.ovab.test-response
```

```
MyMessage
bash-3.2$
```

Exit Status

(Success) – The command displays the message received from the specified vServer.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-reboot-vservers](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-release-ip-addresses

Releases allocated IP addresses of a virtual network.

Syntax

```
iaas-release-ip-addresses|iaas-rip [--base-url <base_url>] [--debug]
                                   [--header] [--help]
                                   [--access-key-file <access_key_file>]
                                   --vnet <vnet-id>
                                   [--ip-addresses] <ip>[,<ip>]*
                                   [--sep <separator>]
                                   [--trust-store <truststore_file>]
                                   [--verbose] [--xml]
```

Description

This command releases the allocated IP addresses of a public or private virtual network. The IP addresses must not be associated with a vServer.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vnet <vnet-id>

vNet ID of the IP addresses to be released.

--ip-addresses <ip>[,<ip>]*

List of IP addresses to be released.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example releases an allocated IP address of the specified virtual network. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vnet	VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c

```
bash-3.2$ ./iaas-release-ip-addresses --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --vnet
VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c --ip-addresses 192.168.0.1
```

```
bash-3.2$
```

Example 2

This example releases three allocated IP addresses of the specified virtual network. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vnet	VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c

```
bash-3.2$ ./iaas-release-ip-addresses --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --vnet
VNET-9634972a-bcdd-4714-8c7f-b67d8010f13c --ip-addresses
192.168.0.2,192.168.0.3,192.168.0.4
```

```
bash-3.2$
```

Exit Status

(Success) – The command releases the allocated IP addresses of the specified virtual network. The command does not return any messages.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-allocate-ip-addresses](#)
- [iaas-create-vnet](#)
- [iaas-describe-ip-addresses](#)

- [iaas-describe-vnets](#)

iaas-run-vserver

Runs a single vServer.

Syntax

```
iaas-run-vserver|iaas-rvs [--base-url <base_url>] [--debug]
                        [--header] [--help]
                        [--access-key-file <access_key_file>]
                        [--dist-group <dg-id>]
                        --ip-addresses <ip>[,<ip>]*
                        [--key-name <key-pair-name>]
                        --name <name>
                        [--desc <descr>]
                        --server-template-id <server-template-id>
                        --vnets <vn-id>[,<vn-id>]*
                        [--volumes <vol-id>[,<vol-id>]*]
                        --vserver-type <type>
                        [--ha <true/false>]
                        [--messages <key>=<value>[,<key>=<value>]*]
                        [--hostname <hostname>]
                        [--root-password <root-password>]
                        [--sep <separator>]
                        [--trust-store <truststore_file>]
                        [--verbose] [--xml]
```

Description

This command creates and starts a single vServer.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--dist-group <dg-id>

A distribution group ID.

--ip-addresses <ip>[,<ip>]*

List of IP addresses, one for each virtual network.

- key-name-l-k <key-pair-name>**
Name of a key pair.
- name | -n <name>**
Name of the vServer.
- desc | -d <descr>**
Description of the vServer.
- server-template-id <server-template-id>**
ID of the server template.
- vnets <vn-id>[,<vn-id>]***
List of virtual network IDs.
- volumes | -m <vol-id>[,<vol-id>]***
List of volume IDs.
- vserver-type <type>]**
vServer type.
- ha <true/false>**
Flag to indicate if HA is enabled or disabled for the vServer. Default value is taken from the HighAvailabilityDefault capability of the vDC.
- messages <key>=<value>[,<key>=<value>]**
List of messages for the vServer.
- hostname <hostname>**
Internal host name for the vServer. It must be RFC 1123 compliant.
- root-password <root-password>**
Root password for the vServer.
- sep <separator>**
Column separator character. The default separator is *TAB*.
- trust-store <truststore_file>**
Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.
- verbose | -v**
Starts the command in verbose mode.
- xml**
Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates and starts a vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

Option	Option Value
key-name	myKeyPair

```
bash-3.2$ ./iaas-run-vserver --base-url https://<EnterpriseControllerHostname>/ -a
~/ak.file --vnets VNET-84ada392-1c13-4f86-8365-1cf7f9c8aadf --key-name myKeyPair
--name myVserver --server-template-id TMPL-9e4a9ed3-e675-45f1-9d7c-b21c25a55632
--ip-addresses 192.168.0.2 --vserver-type 457
```

```
VSRV-0fb57293-347c-4717-96ef-6dd23154596f
bash-3.2$
```

Exit Status

(Success) – The command creates and starts a vServer. This command returns the vServer ID.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-run-vservers

Creates and starts one or more vServers.

Syntax

```
iaas-run-vservers|iaas-rvss [--base-url <base_url>] [--debug]
                             [--header] [--help]
                             [--access-key-file <access_key_file>]
                             [--key-name <key-pair-name>]
                             --name <name>
                             [--desc <descr>]
                             [--dist-group <dg-id>]
                             [--num <num>]
                             --server-template-id <server-template-id>
                             --vnets <vn-id>[,<vn-id>]*
                             [--volumes <vol-id>[,<vol-id>]*]
                             --vserver-type <type>
                             [--ha <true/false>]
                             [--messages <key>=<value>[,<key>=<value>]*]
                             [--hostname <hostname>]
                             [--root-password <root-password>]
                             [--sep <separator>]
                             [--trust-store <truststore_file>]
                             [--verbose] [--xml]
```

Description

This command creates and starts one or more vServers. A suffix is appended to the vServer name to distinguish among all the vServers created. The suffix is a numerical sequence.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax: `https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--key-name | -k <key-pair-name>

Name of the key pair.

--name | -n <name>

Name of the vServer.

--desc | -d <descr>

Description of the vServer.

--dist-group <dg-id>

Distribution group ID.

--num <num>

The number of vServers to run. The default is 1.

--server-template-id <server-template-id>

ID of the server template.

--vnets <vn-id>[,<vn-id>]*

List of virtual network IDs.

--volumes|-m <vol-id>[,<vol-id>]*

List of volume IDs.

--vserver-type <type>]

vServer type.

--ha <true/false>

Flag to indicate whether HA is enabled or disabled for the vServer. Default value is taken from the *HighAvailabilityDefault* capability of the vDC.

--messages <key>=<value>[,<key>=<value>]

List of messages for the vServers.

--hostname <hostname>

Internal host name for the vServers. It must be RFC 1123 compliant.

--root-password <root-password>

Root password for the vServers.

--sep <separator>

Column separator character. The default separator is *TAB*.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

This example creates two vServers. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key-name	myKeyPair

```
bash-3.2$ ./iaas-run-vservers --base-url https://<EnterpriseControllerHostname>/ -a
ak.file --key-name myKeyPair --name myVserver --num 2 --server-template-id
TMPL-2f313208-433c-4b92-aae6-6373c38b795e --vnets
VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4 --vserver-type 457
```

```
VSRV-d6800889-f59b-4798-a57d-3f9f31b0cf1c
VSRV-d6500889-f59b-4567-a65g-3f9f31b0se1d
bash-3.2$
```

Exit Status

(Success) – The command creates and starts the number of specified vServers. This command returns the vServer IDs.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-send-messages-to-vserver

Sends messages to a vServer.

Syntax

```
iaas-send-messages-to-vserver|iaas-smtvs [--base-url <base_url>] [--debug]
                                         [--header] [--help]
                                         [--access-key-file <access_key_file>]
                                         --vserver-id <vserver-id>
                                         [--messages] <key>=<value>[,<key>=<value>]*
                                         [--sep <separator>]
                                         [--trust-store <truststore_file>]
                                         [--verbose] [--xml]
```

Description

This command sends messages to a vServer. These messages can be read from the guest operating system. The message is sent to a common buffer in the virtualization layer that has a limited size. This action might return an error if the size limit is reached or if the vServer is not in an appropriate state to send messages.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vserver-id | -i <vserver-id>

vServer ID.

--messages <key>=<value>[,<key>=<value>]*

List of messages.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example sends a message to a vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vserver-id	VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6

```
bash-3.2$ ./iaas-send-message-to-vserver --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --vserver-id
VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6 --messages com.oracle.ovab.test=Running

bash-3.2$
```

Exit Status

(Success) – The command sends the message to the specified vServer. This command does not return any response, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-start-vservers

Starts one or more vServers for an account.

Syntax

```
iaas-start-vservers|iaas-stavss [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                [--vserver-ids <id>[,<id>]*]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command starts one or more vServers for an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vserver-ids <id>[,<id>]*

List of vServer IDs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

The following example starts the specified vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vserver-id	VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291

```
bash-3.2$ ./iaas-start-vservers --base-url https://<EnterpriseControllerHostname>/  
-a ak.file --vserver-ids VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291
```

```
bash-3.2$
```

Exit Status

(Success) – The command starts the specified vServer. This command does not return any value, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-stop-vservers](#)
- [iaas-terminate-vservers](#)

iaas-stop-vservers

Stops one or more vServers for an account.

Syntax

```
iaas-stop-vservers|iaas-stovss [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                [--vserver-ids <id>[,<id>]*]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command stops one or more vServers for an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the `IAAS_BASE_URL` environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--vserver-ids <id>[,<id>]*

List of vServer IDs.

--sep <separator>

Column separator character. The default separator is `TAB`.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples

Example 1

The following example stops the specified vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vserver-id	VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291

```
bash-3.2$ ./iaas-stop-vservers --base-url https://<EnterpriseControllerHostname>/  
-a ak.file --vserver-ids VSRV-c1e236e6-ef4d-4936-911a-97923dfbc291
```

```
bash-3.2$
```

Exit Status

(Success) – The command stops the specified vServer. This command does not return any response, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-terminate-vservers](#)

iaas-terminate-vservers

Stops and deletes one or more vServers for an account.

Syntax

```
iaas-terminate-vservers|iaas-tvss [--base-url <base_url>] [--debug]
                                [--header] [--help]
                                [--access-key-file <access_key_file>]
                                [--force]
                                [--vserver-ids <id>[,<id>]*]
                                [--sep <separator>]
                                [--trust-store <truststore_file>]
                                [--verbose] [--xml]
```

Description

This command stops and deletes one or more vServers for an account.

Options

--base-url <base_url>

Base URL of the Enterprise Controller where the Web service is running. The value for this option must use the following syntax:

`https://<EnterpriseControllerHostname>`. If this option is not specified, then the URL information is taken from the value set in the IAAS_BASE_URL environment variable.

--debug | -D

Starts the command in debug mode.

--header | -H

Adds a header row to the output. The default output shows no header.

--access-key-file | -a <access_key_file>

Complete path and name of the file that stores the access key (for example, `/tmp/ak.file`). If a path is not specified, then the command looks for the file in the current directory.

--help | -h

Displays the usage information for this command.

--forcelf

Forces the execution of the command.

--vserver-ids <id>[,<id>]*

List of vServer IDs.

--sep <separator>

Column separator character. The default separator is *TAB*.

--trust-store <truststore_file>

Name of the file containing the trusted SSL certificate. The default is `$HOME/.oracle_iaas/truststore`.

--verbose | -v

Starts the command in verbose mode.

--xml

Displays the output result in XML format. The default output is in table format.

Examples**Example 1**

This example deletes a vServer. It uses the following options and values:

Option	Option Value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vserver-id	VSRV-0fb57293-347c-4717-96ef-6dd23154596f

```
bash-3.2$ ./iaas-terminate-vservers --base-url
https://<EnterpriseControllerHostname>/ -a ~/ak.file --vserver-ids
VSRV-0fb57293-347c-4717-96ef-6dd23154596f --force
```

```
bash-3.2$
```

Exit Status

(Success) – The command stops and deletes the specified vServers. This command does not return any response, only the command prompt is returned.

(Failure) – The command displays a descriptive error message and error code.

See Also

- [iaas-describe-vservers](#)
- [iaas-receive-message-from-vserver](#)
- [iaas-run-vserver](#)
- [iaas-run-vservers](#)
- [iaas-send-messages-to-vserver](#)
- [iaas-start-vservers](#)
- [iaas-stop-vservers](#)

Web Service Client Example

This appendix provides an example for generating requests to the Web service of the cloud infrastructure API described in [Chapter 4, "Cloud Infrastructure API Reference"](#).

This example uses the WebUtil application for generating the requests. The appendix includes the following topics:

- [Description of Web Service API Requests](#)
- [Sending a Web Service Request](#)
- [Creating Public and Private Keys](#)
- [Overview of the WebUtil Application](#)
- [Web Service Request Examples](#)

Description of Web Service API Requests

This section provides a description of the information contained in a typical request to the Web service. This section describes both types of requests:

- [AKM Request](#)
- [IAAS Request](#)

AKM Request

The following URL shows the information required in a typical access key management (AKM) request to the Web service:

`https://USER:PASSWORD@HOST/akm/?REQUEST_DATA`

Name	Description
USER	Cloud user name
PASSWORD	Cloud user password
HOST	IP address or fully qualified host name
REQUEST_DATA	The request data is made of a series of name=value parameters separated by an ampersand (&).

REQUEST_DATA Specification

The parameters required as part of the REQUEST_DATA depend on the value used for the AKM action. However, a set of common parameters is required for every AKM request.

The following table shows the required parameters for the REQUEST_DATA in every AKM request. [Example A-1](#) shows how these parameters appear in a complete request.

Name	Description
Action	One of the AKM actions. See RegisterAccessKeyRequest , DescribeAccessKeysRequest , DeleteAccessKeyRequest , and DescribeAccountsRequest actions for a complete description and extra parameters for each AKM action.
Version	1
Timestamp	Timestamp in milliseconds since January 1, 1970
Expires	Expiry date in milliseconds (ms) since January 1, 1970 must be greater than the timestamp field. A 300000 ms difference between the two values must be enough.

Example A-1 A Complete AKM Request

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1330954919299
```

IAAS Request

The following URL shows the information required in a typical infrastructure as a service (IAAS) request to the Web service:

```
https://HOST/iaas/?REQUEST_IAAS_DATA&SIGNATURE_BLOCK
```

Name	Description
HOST	IP address or fully qualified host name
REQUEST_IAAS_DATA	The request IAAS data is a series of name=value parameters separated by an ampersand (&).
SIGNATURE_BLOCK	The signature block is a series of name=value parameters separated by an ampersand (&).

REQUEST_IAAS_DATA Specification

The parameters required as part of the REQUEST_IAAS_DATA depend on the value used for the IAAS action. However, a set of common parameters is required for every IAAS request.

The following table shows the common required parameters for the REQUEST_IAAS_DATA in every IAAS request. [Example A-2](#) show how these parameter appears in a complete specification for REQUEST_IAAS_DATA.

Name	Description
Action	One of the IAAS actions. See " List of Actions " on page on page 4-4, for a complete list of IAAS actions and extra parameters for each IAAS action.
Version	1

Name	Description
Timestamp	Timestamp in milliseconds since January 1, 1970
Expires	Expiry date in milliseconds (ms) since January 1, 1970 must be greater than the timestamp field. A 300000 ms difference between the two values must be enough.
accessKeyId	The value returned by the AKM RegisterAccessKeyRequest action.

Example A-2 Sample REQUEST_IAAS_DATA Specification

```
Action=
DescribeVnets&Version=1&Timestamp=1330954619299&Expires=1330954919299&accessKeyId=
AK_1
```

SIGNATURE_BLOCK Specification

The signature block consists of signing some IAAS data with a private key so that the Web service can authenticate the request. The Web server should be able to verify the signature with the public key registered with the [RegisterAccessKeyRequest](#) action.

The following table shows the series of name=value parameters that conform the specification for the SIGNATURE_BLOCK. These parameters are separated by the ampersand(&). [Example A-3](#) shows a complete specification for a SIGNATURE_BLOCK sample.

Name	Description
SignatureVersion	1
SignatureMethod	SHA512withRSA. This is the only method supported.
Signature	Encoded Hash value of the data to be signed. See " IAAS Data Used for Signature " for a complete description of the Signature field.

Example A-3 Sample SIGNATURE_BLOCK Specification

```
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=bj8GfJCqvPZZPU2JoWAGzZd
CF+N767rQejILMQwNdgKLfoGGqAwDPRYMr/ghUoBc6RB3nKYgAyPdmtCfhzRGTqECgUWy0jCrE99+utGee
J0/XRQ9LxyYeBgZjO3lHP+hFhUo+gUtQaSYPhUHH7eTkxg/Cro1Mxibgl1ypJM/rIf90yEqSeqhphQt7hWx
lT0DNAy6/cZt8isT/Tu8V7ZFjBfkEpLfn97bIOJ2vIIP0eetmftuw40btqjbUp6+7dpVkhkCQnX0MAIDj+
mJOREOzCwK+F1pYuzES0fjaW0MowG+cA/9gttDjg7r5H29i3qbbjI1vAt6fk1HPpSxQTSTOTg==
```

IAAS Data Used for Signature

The following table shows the required information to generate the encoded hash value of the signature for an IAAS request. This data needs to be signed and base64 encoded. [Example A-4](#) shows a complete data set of the IAAS data used for generating a signature. [Example A-5](#) shows a Java representation of the IAAS data to sign.

Name	Description
Http Request type	POST
HostName	IP address or fully qualified host name
Base URL of the Web service	iaas

Name	Description
REQUEST_IAAS_DATA	The request data is made of a series of name=value parameters separated by an ampersand (&). See also "REQUEST_IAAS_DATA Specification" .

Example A-4 Example of the IAAS Data to Sign

A "\n" is required between every field.

```
POST
<EnterpriseControllerHostname>
/iaas/
Action= DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1330954919299
```

Example A-5 Java Representation of the Data to Sign

```
StringBuilder message = new StringBuilder();
Message.setLength(0);
message.append("POST").append("\n");
message.append("<EnterpriseControllerHostname>") .append("\n");
message.append("/iaas/").append("\n");
message.append("Action=
DescribeVnets&Version=1&Timestamp=1330954619299&Expires=1330954919299&accessKeyId=
AK_1
").append("\n");
```

See ["Web Service Request Examples"](#), for an example of generating an IAAS requests with signed data.

Sending a Web Service Request

Requests to the Web service are simple HTTP POST or GET operations. After a AKM or IAAS request is dynamically generated, static technology is used to send the request.

This section mentions the use of the Web browser and the WGET utility as the static technology to send requests to the Web service.

Using the Web Browser

To use the Web browser to send a request to the Web service, enter the complete request in the address bar of the browser.

The following URL is an example of a complete AKM request. The request should be entered in the address bar as a one-line command:

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/? Action=
DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1330954919299
```

Using WGET Utility

To use WGET utility, it is necessary to use the `--no-check-certificate` option.

The following is an example of sending a request using WGET with the same AKM request used previously:

```
wget https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=
DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1330954919299
```

```
--nocheck-certificate
```

Creating Public and Private Keys

Both public and private RSA keys are required for signing the required IAAS data so that the Web service can authenticate the request.

Private Key

A private key must exist before you can create a public key. To create a private key use the following command:

```
openssl genrsa -out privatekey.pem 2048
```

The private key is stored in the privatekey.pem file. This file is used to create the public key. After the public key is registered with the [RegisterAccessKeyRequest](#) action, this private key can be used to sign the IAAS data.

If the WebUtil application is used to sign the IAAS data, then the private key must be stored in DER format. You can use the following command:

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in privatekey.pem -nocrypt > privatekey.DER
```

The private key is stored in the privatekey.DER file.

Public Key

To create a public key use the following command:

```
openssl rsa -in privatekey.pem -pubout -out publickey.pem
```

The public key is stored in the publickey.pem file. After the public key is registered using the [RegisterAccessKeyRequest](#) action, the Web server can verify the signed data with the public key.

Overview of the WebUtil Application

WebUtil is a simple Java application that generates the signature for an IAAS request to the Web service. WebUtil uses the SHA512withRSA method and base64 encoding to sign the required data.

The following is the code of the WebUtil application. You can build the application with the javac compiler; it is possible that two warnings are displayed when building the application.

```
/*
 * Copyright (c) 2007, 2012 Oracle and/or its affiliates. All rights reserved.
 * Use is subject to license terms.
 */
import java.io.*;
import java.nio.charset.Charset;
import java.security.*;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.interfaces.RSAPrivateKey;
public class WebUtil {
    final static String UTF_8_ENCODER = "UTF-8";
    private static final Charset CHARSET_ENCODING_UTF_8 = Charset.forName("UTF-8");
    private static final long IAAS_REQUEST_TIMEOUT_MS = 300000; //default expiry
    time is 5 minutes
```

```
public static void main(String[] args) {
    if(args.length<=0) {
WebUtil.usage();
    }
    String argument=new String(args[0]);
    if(argument.compareTo("signature") ==0 && args.length != 6){
WebUtil.usage();
    }
    if(argument.compareTo("template") == 0 && args.length != 2){
WebUtil.usage();
    }

    try{

        if(argument.compareTo("template") == 0){
WebUtil.generateTemplate(args[1]);

        }

        if(argument.compareTo("signature") ==0 && args.length == 6){
WebUtil.signDataToFile(args[1],args[2],args[3],args[4],args[5]);

        }

    }catch (Exception e){
        System.err.println("Caught exception " + e.toString());
    }

}

static void usage(){

        System.out.println("Usage: WebUtil template destinationFile");
        System.out.println("Usage: WebUtil signature privateKeyDER HTTP_TYPE HOST_IP
DATA_TO_SIGN signatureData");
        System.out.println("Usage: HTTP_TYPE: POST or GET\n HOST_IP=ip address of OC
\n DATA_TO_SIGN:without Timestamp and Expire but needs to have the access key
id \n signatureData:filename to store the signature");
        System.exit(0);

}

static void generateTemplate(String filename)throws Exception {
    StringBuilder message = new StringBuilder();

    message.setLength(0);
    message.append("POST").append("\n");
    message.append("10.169.79.76").append("\n");
    message.append("/iaas/").append("\n");
    message.append("Action=DescribeVnets&Version=1");
    message.append("&Timestamp=1330797956376");
    message.append("&Expires=1330798256376");
    message.append("&AccessKeyId=AK_1");
    message.append("\n");
    FileOutputStream iaas = new FileOutputStream(filename);
        iaas.write(message.toString().getBytes(CHARSET_ENCODING_UTF_8));
        iaas.close();
    }
}
```

```

static void signDataToFile(String keyPrivFile, String httpType, String host,
String dataRequest,String signedDataFile) throws Exception {

    long tsnow = System.currentTimeMillis();
    long tsexpires = tsnow + IAAS_REQUEST_TIMEOUT_MS;
    String tNow=Long.toString(tsnow);
    String tEXPIRE=Long.toString(tsexpires);

    StringBuilder message = new StringBuilder();
    message.setLength(0);
    message.append(httpType).append("\n");
    message.append(host).append("\n");
    message.append("/iaas/").append("\n");
    message.append(dataRequest);
    message.append("&Timestamp="); message.append(tNow);
    message.append("&Expires="); message.append(tEXPIRE);
    message.append("\n");

    StringBuilder iaasmessage = new StringBuilder();
    iaasmessage.append("https://").append(host);
    iaasmessage.append("/iaas/?").append(dataRequest);
    iaasmessage.append("&Timestamp="); iaasmessage.append(tNow);
    iaasmessage.append("&Expires="); iaasmessage.append(tEXPIRE);
    iaasmessage.append("&SignatureMethod=SHA512withRSA&SignatureVersion=1");
    iaasmessage.append("&Signature=");

    String messageStr = message.toString();

    /* Read private keyfile DER */
    FileInputStream in = new FileInputStream( keyPrivFile);

    FileInputStream keyfis =    new FileInputStream(keyPrivFile);
    byte[] encKey = new byte[keyfis.available()];
    keyfis.read(encKey);
    keyfis.close();

    PKCS8EncodedKeySpec privKeySpec = new PKCS8EncodedKeySpec(encKey);

    KeyFactory keyFactory =    KeyFactory.getInstance("RSA");
    PrivateKey privKey =    keyFactory.generatePrivate(privKeySpec);

    Signature rsa = Signature.getInstance("SHA512withRSA");

        rsa.initSign(privKey);
        rsa.update(messageStr.getBytes(CHARSET_ENCODING_UTF_8));

    /* Now that all the data to be signed has been read in,
        generate a signature for it */

        byte[] realSig = rsa.sign();

    /* base64 encode the signed data */
    String signatureUrlEncoded = getBase64Encoder().encode(realSig);

```

```
/* Copy data to file */

FileOutputStream iaas = new FileOutputStream(signedDataFile);
    iaas.write(signatureUrlEncoded.getBytes(CHARSET_ENCODING_UTF_8));
    iaas.close();

/* copy part of request to request file */
FileOutputStream iaasreq = new FileOutputStream("iaasPartRequest");
    iaasreq.write(iaasmessage.toString().getBytes(CHARSET_ENCODING_UTF_
8));
    iaasreq.close();

}

/**
 * @return a new base 64 encoder
 */
public static sun.misc.BASE64Encoder getBase64Encoder() {
    /*
     * This helper method was introduced to minimize the warnings about
     * using a Sun proprietary API. Use full package names to avoid
     * warning on import statement.
     */
    return new sun.misc.BASE64Encoder();
}
}
```

How the WebUtil Application Works

WebUtil application requires an access key for the account. See ["Creating an Access Key"](#) for information about how to create the access key.

After the access key is created, the IAAS data must be defined, as explained in ["IAAS Data Used for Signature"](#).

The following is an example of the IAAS data for viewing vNets information:

```
POST
<EnterpriseControllerHostname>
Action=DescribeVnets&Version=1&accessKeyId=AK_
1&Timestamp=12333333&Expires=13333444
```

To generate the signature and base64 encoding, use the WebUtil by running the following command:

```
java WebUtil signature privatekey.DER "POST" "<EnterpriseControllerHostname>"
"Action=DescribeVnets&Version=1&accessKeyId=AK_1" signedData
```

WebUtil generates and stores the signed data in the signedData file:

Content of the signedData file:

```
bj8GfJCqvPZZPU2JoWAGzZdCF+N767rQejILMQwNdgKLfoGGqAwDPRYMr/ghUoBc6RB3nKYgAy
PdmTcfhzRGTqECgUWy0jCrE99+utGeeJ0/XRQ9LxyYeBgZj03lHP+hFhUo+gUtQaSYPhUHH7eT
kxg/CrolMxibgIypJM/rIf90yEqSeqhphQt7hWx1T0DNay6/cZt8isT/Tu8V7ZFjBFkEpLfn97
bIOJ2vIIPoetmftuw40btqjBUp6+7dpVkhCQnX0MAIDj+mjorEOzcwK+F1pYuzES0fjaW0Mo
wG+cA/9gttDjg7r5H29i3qbbjIlvAt6fk1HPpSxQTSTOTg==
```

WebUtil also generates the iaasPartRequest file with the following information:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Version=
1&AccessKeyId=AK_
1&Timestamp=1331058169938&Expires=1331058469938&SignatureMethod=SHA512with
RSA&SignatureVersion=1&Signature=
```

To generate the complete HTTP request, append the content of the signedData file to the iaasPartRequest file as a single-line command. After that, the HTTP request can be sent, see ["Sending a Web Service Request"](#).

Web Service Request Examples

This section presents an example for creating requests to the Web service that allows a cloud user to perform actions for an account. This section includes:

- [Creating an Access Key](#)
- [Executing actions for an account](#)

Creating an Access Key

An access key is created for an account. The registered access key is needed to perform any IAAS action in the account.

For this example, the following information is used to create the access key:

- USER: *clouduser*
- PASSWORD: *<password>*
- HOST: *10.16.79.70*

To create an access key:

1. Create a private RSA key stored as a DER file.
 - a. Run the following command to create a private RSA key file:


```
openssl genrsa -out privatekey.pem 2048
```
 - b. Run the following command to store the private key as DER:


```
openssl pcks8 -topk8 -inform PEM -outform DER -in privatekey.pem
-nocrypt > privatekey.DER
```
2. Create a public RSA key base on the privateRSA key. Run the following command:

```
openssl rsa -in privatekey.pem -pubout -out publickey.pem
```

The openssl public key generated must be similar to the following:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEaufVdjdP0MmOLbNypLVMW
Xfmhusawid4Wg4n4FZewSmoBEYA8f8wIA0SI87Shi7RtMcWsEoXvNNHA0wcJoA1R
jyVLsI3rtrq0c0k7AxQSwb4UK/rSXW1NXxMh/mE7b3gdA6d9VuwIPnZJ5ZFUZCL
yhaAotLCdAcRzbgzYXdqt+rstutT1AVkE2UAMcm503KnIoObZKb8JtepSt74A9Rg
VBkcCBjmKGfLNOL1K1ZconkITm85TWKRAGRfuASxd12ZrD723ZNb66X/a9ebxTMr
6vVeskcaZpPlHzvgMOpYDGwRvxn9yM5WB83zFDGT26LihN/bKzLjXa+F2YNkLrT
JQIDAQAB
-----END PUBLIC KEY-----
```

3. Display the account attributes for a cloud user.

- a. Create and send an AKM request using the [DescribeAccountsRequest](#) action. The appropriate hostname, cloud user, timestamp, and expires information must be provided:

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?
Action=DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1
330954919299
```

- b. Verify the account ID from the result returned from the previous AKM request. In this example, the account ID is ACC-0162da5a-5d25-4096-af59-3dd1de27cfad

```
<result xsi:type="DescribeAccountsResult" requestId="803">
<items>
<account>ACC-0162da5a-5d25-4096-af59-3dd1de27cfad</account>
<name>acc1</name>
</items>
<forUser><username></forUser>
</result>
```

4. Create the access key for a cloud user account.

- a. Register the public key created in step 2 by using the [RegisterAccessKey](#) action. Ensure that the data is represented as a single line. The following URL shows the HTTPS request to create an access key. The account used is the one from the [DescribeAccount](#) action used in step 3:

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?A
ction=RegisterAccessKey&Version=1&Timestamp=1330975344&Expires=1333
975344&account=ACC-0162da5a-5d25-4096-af59-3dd1de27cfad&publicKey=M
IIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEufVdjdP0MmOLbNypLVMWXfmh
usawid4Wg4n4FZewSmoBEYA8f8wIA0SI87Shi7RtMcWseOxvNNHA0wcJoA1RjyVLsI3
rtrq0c0k7AxQSwb4UK/rSXW1NXxMh/mE7b3gdA6d9VuwIPnZJ5ZFQUZCLyhaAotLCdA
CrzbgzYXdqt+rstutT1AVkE2UAMcm503KnIoObZKb8JtepSt74A9RgVBkcCBjmKGFln
OL1K1ZconkITm85TWKRAGRFuASxd12ZrD723ZNb66X/a9ebxTMr6vVeskcaZpPlHzvg
MOpiyDGwRvxn9yM5WB83zFDGT26Lihn/bKzLJXa+F2YNkLrTJQIDAQAB
```

- b. Verify the access key ID that was generated. The access key ID is part of the results returned from the [RegisterAccessKey](#) action. This access key ID is used for all IAAS requests for the account to authenticate the cloud user to the account. In this examples, the access key ID is AK_1.

```
<result xsi:type="RegisterAccessKeyResult" requestId="1013">
<accessKeyId>AK_1</accessKeyId>
</result>
```

Executing actions for an account

The following actions are explained in this section:

- [Viewing All Available Virtual Networks](#)
- [Viewing Specific Virtual Networks by Using IDs](#)
- [Viewing Specific Virtual Networks by Using Filters](#)
- [Creating a Private Virtual Network](#)

To execute these actions using the Web service, the access key ID and private key file obtained from the previous step.

Viewing All Available Virtual Networks

To view attributes of all available virtual networks for an account:

1. Define the data to be signed by using the DescribeVnets action.

```
Action=DescribeVnets&Version=1&AccessKeyId=AK_1
```

2. Sign the data by using the WebUtil application.

```
java WebUtil signature privatekey.DER "POST" "<EnterpriseControllerHostname>"
"Action=DescribeVnets&Version=1&accessKeyId=AK_1" signedData
```

3. Get the signed data that is stored in the iaasPartRequest and signedData files.

Content of the iaasPartRequest file:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Version=1&AccessKeyId=AK_1&Timestamp=1331058169938&Expires=1331058469938&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=
```

Content of the signedData file:

```
bj8GfJCqvPZZPU2JoWAGzZdCF+N767rQejILMQwNdGKLfoGGqAwDPRYMr/ghUoBc6RB3nKYgAyPdmtCfhzRGTqECgUWy0jCrE99+utGeeJ0/XRQ9LxyYeBgZj03lHP+hFhUo+gUtQaSYPhUHH7eTkxg/CrolMxibglypJM/rIf90yEqSeqhphQt7hWx1T0DNAY6/cZt8isT/Tu8V7ZFjBFkEpLfn97bIOJ2vIIpOeetmftuw4ObtqjbUp6+7dpVkhkCQnX0MAIDj+mjorEOzck+F1pYuzES0fjaW0MowG+cA/9gttDjg7r5H29i3qbbjI1vAt6fk1HPPsXQTSTOTg==
```

4. Append the signature to get the complete IAAS request, as show in the following example:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Version=1&AccessKeyId=AK_1&Timestamp=1331058169938&Expires=1331058469938&SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=bj8GfJCqvPZZPU2JoWAGzZdCF+N767rQejILMQwNdGKLfoGGqAwDPRYMr/ghUoBc6RB3nKYgAyPdmtCfhzRGTqECgUWy0jCrE99+utGeeJ0/XRQ9LxyYeBgZj03lHP+hFhUo+gUtQaSYPhUHH7eTkxg/CrolMxibglypJM/rIf90yEqSeqhphQt7hWx1T0DNAY6/cZt8isT/Tu8V7ZFjBFkEpLfn97bIOJ2vIIpOeetmftuw4ObtqjbUp6+7dpVkhkCQnX0MAIDj+mjorEOzck+F1pYuzES0fjaW0MowG+cA/9gttDjg7r5H29i3qbbjI1vAt6fk1HPPsXQTSTOTg==
```

Note: The complete request to the Web service must be created as a single-line request.

5. Send the complete IAAS request using the browser or the WGET utility. The result returned is similar to the following output:

```
<result xsi:type="DescribeVnetsResult" requestId="102">
<items>
<id>VNET-6c774d08-d0a7-493b-9fa1-c93a80702f8d</id>
<name>10.169.69.0/24</name>
<description>OVM Discovered Network</description>
<status>OK</status>
<ipAddress>10.169.69.0/24</ipAddress>
</items>
</result>
```

Viewing Specific Virtual Networks by Using IDs

To use the virtual networks IDs to specify which virtual networks to display for an account:

1. Define the data to be signed by using the DescribeVnets action.

```
Action=DescribeVnets&Version=1&AccessKeyId=AK_
1&ids.1=VNET-6c774d08-d0a7-493b-9fa1-c93a80702f8d
```

2. Sign the data by using the WebUtil application.

```
java WebUtil signature privatekey.DER "POST" "<EnterpriseControllerHostname>"
"Action=DescribeVnets&Version=1&AccessKeyId=AK_
1&ids.1=VNET-6c774d08-d0a7-493b-9fa1-c93a80702f8d" signedData
```

3. Get the signed data that is stored in the iaasPartRequest and signedData files.

Content of the iaasPartRequest file:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Versi
on=1&AccessKeyId=AK_
1&ids.1=VNET-6c774d08-d0a7-493b-9fa1-c93a80702f8d&Timestamp=13310584499
00&Expires=1331058749900&SignatureMethod=SHA512withRSA&SignatureVersion
=1&Signature=
```

Content of the signedData file:

```
B15A5WdQkIwTr1R/NDXbn1RFwWYBFAQFtZTfz1jH8ftkRgvfwU93nxukpw23sHBdRvzhLuh
yW2LWm90EIhZHd3H2x9s8D1wCAUFTsSAZoPuI1W40qn0sf9VTEXOlqwnxfray6eSGCcpHQW
XaA+TjpCitZguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmbH7iI7pBC8gcpomnDZhx
ZkAFaVT0TFf88epbJa15Y16aTpptj1UNEz6jDhuV2Qaa3bI50qnJc0PJgCkhuKU7un8us1J
VEk/nf/or5RCBMmPGAluuILIfYBEe/euVUy7wQ00eyKPRup1ZkORmVbFLg==
```

4. Append the signature to get the complete IAAS request, as show in the following example:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Versi
on=1&AccessKeyId=AK_
1&ids.1=VNET-6c774d08-d0a7-493b-9fa1-c93a80702f8d&Timestamp=13310584499
00&Expires=1331058749900&SignatureMethod=SHA512withRSA&SignatureVersion
=1&Signature=B15A5WdQkIwTr1R/NDXbn1RFwWYBFAQFtZTfz1jH8ftkRgvfwU93nxukpw
23sHBdRvzhLuhyW2LWm90EIhZHd3H2x9s8D1wCAUFTsSAZoPuI1W40qn0sf9VTEXOlqwnxf
ray6eSGCcpHQWxXaA+TjpCitZguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmbH7iI7p
BC8gcpomnDZhxZkAFaVT0TFf88epbJa15Y16aTpptj1UNEz6jDhuV2Qaa3bI50qnJc0PJgC
khuKU7un8us1JVEk/nf/or5RCBMmPGAluuILIfYBEe/euVUy7wQ00eyKPRup1ZkORmVbFLg
==
```

Note: The complete request to the Web service must be created as a single-line request.

5. Send the complete IAAS request using the browser or WGET utility. The result returned is similar to the following output:

```
<result xsi:type="DescribeVnetsResult" requestId="107">
<items>
<id>VNET-6c774d08-d0a7-493b-9fa1-c93a80702f8d</id>
<name>10.169.69.0/24</name>
<description>OVM Discovered Network</description>
<status>OK</status>
<ipAddress>10.169.69.0/24</ipAddress>
</items>
```

```
</result>
```

Viewing Specific Virtual Networks by Using Filters

To specify which virtual networks to display by using the name of the virtual network as filter:

1. Define the data to be signed by using the DescribeVnets action:

```
Action=DescribeVnets&Version=1&AccessKeyId=AK_
1&filters.1.filterValue=privatevnet1&filters.1.filterName=name
```

2. Sign the data by using the WebUtil application.

```
java WebUtil signature privatekey.DER "POST" "<EnterpriseControllerHostname>"
"Action=DescribeVnets&Version=1&AccessKeyId=AK_
1&filters.1.filterValue=privatevnet1&filters.1.filterName=name" signedData
```

3. Get the signed data that is stored in the iaasPartRequest and signedData files.

Content of the iaasPartRequest file:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Versi
on=1&AccessKeyId=AK_
1&filters.1.filterValue=privatevnet1&filters.1.filterName=name&Timestam
p=1331058887344&Expires=1331059187344&SignatureMethod=SHA512withRSA&Sig
natureVersion=1&Signature=
```

Content of the signedData file:

```
B15A5WdQkIwTrlR/NDXbn1RFwWYBFAQFtZTfz1jH8ftkRgvfwU93nxukpw23sHBdRvhzLuh
yW2LWm90EIhZHd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxfray6eSGCcpHQW
XaA+TjpCitZguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmbH7iI7pBC8gcpomnDZhx
ZkAFaVT0TFf88epbJa15Yl6aTpptj1UNEz6jDhuV2Qaa3bI50qnJc0PJgCkhuKU7un8us1J
VEk/nf/or5RCBMmPGAluuILIfYBEE/euVUY7wQ00eyKPRuplZkORmVbFLg==
```

4. Append the signature to get the complete IAAS request, as show in the following example:

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=
1&AccessKeyId=AK_
1&name=privatevnetWebApi&Timestamp=1331058639019&Expires=1331058939019&
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=B15A5WdQkIwTr
rlR/NDXbn1RFwWYBFAQFtZTfz1jH8ftkRgvfwU93nxukpw23sHBdRvhzLuhyW2LWm90EIhZ
Hd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxfray6eSGCcpHQWXAa+TjpCitZg
uoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmbH7iI7pBC8gcpomnDZhxZkAFaVT0TFf8
8epbJa15Yl6aTpptj1UNEz6jDhuV2Qaa3bI50qnJc0PJgCkhuKU7un8us1JVEk/nf/or5RC
BMmPGAluuILIfYBEE/euVUY7wQ00eyKPRuplZkORmVbFLg==
```

Note: The complete request to the Web service must be created as a single-line request.

5. Send the complete IAAS request using the browser or wget. The result returned is similar to the following output:

```
<result xsi:type="DescribeVnetsResult" requestId="111">
<items>
<id>VNET-8028fbfa-9e6f-4494-82c5-b35367340240</id>
<name>privatevnet1</name>
<description>privatevnet1</description>
<status>OK</status>
<ipAddress>192.168.0.0/24</ipAddress>
```

```
</items>
</result>
```

Creating a Private Virtual Network

To create a private virtual network for an account:

1. Define the data to be signed by the using DescribeVnets action.

```
Action=CreateVnet&Version=1&AccessKeyId=AK_1&name=privatevnetWebApi
```

2. Sign the data by using the WebUtil application.

```
java WebUtil signature privatekey.DER "POST" "<EnterpriseControllerHostname>"
"Action=CreateVnet&Version=1&AccessKeyId=AK_1&name=privatevnetWebApi"
signedData
```

3. Get the signed data that is stored in the iaasPartRequest and signedData files.

Content of the iaasPartRequest file:

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=
1&AccessKeyId=AK_
1&name=privatevnetWebApi&Timestamp=1331058639019&Expires=1331058939019&
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=
```

Content of the signedData file:

```
B15A5WdQkIwTrlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw23sHBdRvzhLuh
yW2LWm90EihZHd3H2x9s8D1wCAUFTsSAZoPuI1W40qn0sf9VTEXOlqwnxfray6eSGCcpHQW
XaA+TjpCitZguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzMH7iI7pBC8gcpomnDZhx
ZkAFaVT0TFf88epbJa15Y16aTpptj1UNEz6jDhuV2Qaa3bI50qnJc0PJgCkhuKU7un8us1J
VEk/nf/or5RCBMmPGAluuILIfYBEe/euVUy7wQ00eyKPRup1ZkORmVbFLg==
```

4. Append the signature to get the complete IAAS request, as show in the following example:

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=
1&AccessKeyId=AK_
1&name=privatevnetWebApi&Timestamp=1331058639019&Expires=1331058939019&
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=B15A5WdQkIwT
rlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw23sHBdRvzhLuhyW2LWm90EihZ
Hd3H2x9s8D1wCAUFTsSAZoPuI1W40qn0sf9VTEXOlqwnxfray6eSGCcpHQWXaA+TjpCitZg
uoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzMH7iI7pBC8gcpomnDZhxZkAFaVT0TFf8
8epbJa15Y16aTpptj1UNEz6jDhuV2Qaa3bI50qnJc0PJgCkhuKU7un8us1JVEk/nf/or5RC
BMmPGAluuILIfYBEe/euVUy7wQ00eyKPRup1ZkORmVbFLg==
```

Note: The complete request to the Web service must be created as a single-line request.

5. Send the complete IAAS request using the browser or WGET utility. The result returned is similar to the following output:

```
<result xsi:type="CreateVnetResult" requestId="109">
<vnetId>VNET-7e33ee2e-d0f9-4b39-b8c9-32b8c8bfed87</vnetId>
</result>
```

Index

A

Access key management, 1-4, 3-3

- CLI commands
 - akm-create-access-key, 3-4, 5-4
 - akm-delete-access-key, 3-5, 5-7
 - akm-describe-access-keys, 3-4, 5-10
 - akm-describe-accounts, 5-13
- Cloud infrastructure API actions
 - CreateAccessKeyAsObjectRequest, 3-4, 4-9
 - CreateAccessKeyToFileRequest, 4-10
 - DeleteAccessKeyRequest, 3-5, 4-24
 - DescribeAccessKeysRequest, 3-4, 4-33
 - DescribeAccountsRequest, 4-35
 - RegisterAccessKeyRequest, 3-4, 4-74
- creation, 3-4
- deletion, 3-5
- viewing information, 3-4

Access keys, 3-1

- creation, 3-4
- delete, 3-5
- viewing information, 3-4

Account management, 1-3, 3-1

Accounts, 1-2

- access management, 1-4, 3-1, 3-3
- management, 1-3
- resource attributes, 3-1, 3-28, 3-29
- resource status, 3-2
- resources, 1-1, 3-1

akm-create-access-key command, 3-4, 5-4

akm-delete-access-key command, 3-5, 5-7

akm-describe-access-keys command, 3-4, 5-10

akm-describe-accounts command, 5-13

AllocateIpAddressesRequest action, 3-8, 4-5

APIs and CLI

- execution flow, 1-2
- installation, 2-1
- prerequisites, 2-1

assembly, 3-11

AttachVolumesToVserverRequest action, 3-15, 4-7

Attributes, 3-1

- CLI commands
 - iaas-describe-attributes, 3-28, 5-57
 - iaas-modify-attributes, 3-29, 5-97
- Cloud infrastructure API actions
 - DescribeAttributesRequest, 3-28, 4-37

- ModifyAttributesRequest, 3-29, 4-69
- distribution groups, 3-19
- management, 3-28
- modify, 3-29
- snapshots, 3-17
- view, 3-28
- virtual networks, 3-7
- volumes, 3-14
- vServer types, 3-20
- vServers, 3-22, 3-23

C

CLI, 1-1, 1-2

- common command options, 5-1
- enterprise controller, 2-2
- installation, 2-2
- list of commands, 5-3
- package, 2-2
- prerequisites, 2-1

Cloud administrators, 1-2

Cloud infrastructure API, 1-1, 1-2

- data types, 4-98
- enterprise controller, 2-1
- installation, 2-1
- list of actions, 4-4
- overview, 4-1
- prerequisites, 2-1
- request examples, 4-2
- request parameters, 4-2
- request signature, 4-2
- requests, 4-1

Cloud infrastructure CLI

- See CLI, 1-1

Cloud infrastructure Java API, 1-1, 1-2

- installation, 2-1
- package, 2-1
- prerequisites, 2-1

Cloud user, 1-2

- management, 1-1

CreateAccessKeyAsObjectRequest action, 3-4, 4-9

CreateAccessKeyToFileRequest action, 4-10

CreateDistributionGroupRequest action, 3-18, 4-12

CreateKeyPairAsObjectRequest action, 4-14

CreateKeyPairToFileRequest action, 4-15

CreateSnapshotRequest action, 3-17, 4-16

CreateTagsRequest action, 3-30, 4-18
CreateVnetRequest action, 3-6, 4-20
CreateVolumeRequest action, 3-13, 4-22

D

Data types, 4-98
DeleteAccessKeyRequest action, 3-5, 4-24
DeleteDistributionGroupRequest action, 3-19, 4-25
DeleteKeyPairRequest action, 3-27, 4-26
DeleteSnapshotRequest action, 3-17, 4-27
DeleteTagsRequest action, 4-28
DeleteVnetRequest action, 3-7, 4-30
DeleteVolumeRequest action, 3-14, 4-31
DeregisterServerTemplateRequest action, 3-12, 4-32
DescribeAccessKeysRequest action, 3-4, 4-33
DescribeAccountsRequest action, 4-35
DescribeAttributesRequest action, 3-28, 4-37
DescribeDistributionGroupsRequest action, 3-19, 4-39
DescribeIpAddressesRequest action, 3-9, 4-41
DescribeKeyPairsRequest action, 3-28, 4-43
DescribeServerTemplatesRequest action, 3-12, 4-45
DescribeSnapshotsRequest action, 3-18, 4-47
DescribeTagsRequest action, 3-30, 4-49
DescribeVdcCapabilitiesRequest action, 3-5, 4-51
DescribeVnetsRequest action, 3-7, 4-53
DescribeVolumesRequest action, 3-14, 4-55
DescribeVserverMetricsRequest action, 3-24, 4-57
DescribeVserversRequest action, 3-23, 4-59
DescribeVserverTypesRequest action, 3-21, 4-62
DetachVolumesFromVserverRequest action, 3-15, 4-63
Distribution group management, 1-5, 3-18
 CLI commands
 iaas-create-distribution-group, 3-18, 5-21
 iaas-delete-distribution-group, 3-19, 5-43
 iaas-describe-distribution-groups, 3-19, 5-60
Cloud infrastructure API actions
 CreateDistributionGroupRequest, 3-18, 4-12
 DeleteDistributionGroupRequest, 3-19, 4-25
 DescribeDistributionGroupsRequest, 3-19, 4-39
 creating, 3-18
 deleting, 3-19
 viewing information, 3-19
Distribution groups, 3-18
 attributes, 3-19
 create, 3-18
 delete, 3-19
 management, 1-5
 view information, 3-19

E

Enterprise controller, 2-1, 2-2

H

High availability, 3-21, 3-29

I

iaaS, 1-1
 APIs, 1-1
 APIs and CLI execution flow, 1-2
 CLI, 1-1
iaas-allocate-ip-addresses command, 3-8, 5-16
iaas-attach-volumes-to-vserver command, 3-15, 5-19
iaas-create-distribution-group command, 3-18, 5-21
iaas-create-key-pair command, 3-26, 5-24
iaas-create-server-template-from-assembly
 command, 3-11, 5-26
iaas-create-server-template-from-url
 command, 3-11, 5-28
iaas-create-server-template-from-vserver
 command, 3-10, 5-30
iaas-create-snapshot command, 3-17, 5-32
iaas-create-tags command, 3-30, 5-35
iaas-create-vnet command, 3-6, 5-37
iaas-create-volume command, 3-13, 5-40
iaas-delete-distribution-group command, 3-19, 5-43
iaas-delete-key-pair command, 3-27, 5-45
iaas-delete-server-template command, 3-12, 5-47
iaas-delete-snapshot command, 3-17, 5-49
iaas-delete-tags command, 3-30, 5-51
iaas-delete-vnet command, 3-7, 5-53
iaas-delete-volume command, 3-14, 5-55
iaas-describe-attributes command, 3-28, 5-57
iaas-describe-distribution-groups command, 3-19, 5-60
iaas-describe-ip-addresses command, 3-9, 5-62
iaas-describe-key-pairs command, 3-28, 5-64
iaas-describe-server-templates command, 3-12, 5-67
iaas-describe-snapshots command, 3-18, 5-70
iaas-describe-tags command, 3-30, 5-72
iaas-describe-vdc-capabilities command, 3-5, 5-75
iaas-describe-vnets command, 3-7, 5-77
iaas-describe-volumes command, 3-14, 5-80
iaas-describe-vserver-metrics command, 3-24, 5-83
iaas-describe-vservers command, 3-23, 5-86
iaas-describe-vserver-types command, 3-21, 5-89
iaas-detach-volumes-from-vserver command, 3-15, 5-91
iaas-import-key-pair command, 3-27, 5-93
iaas-import-volume command, 3-16, 5-95
iaas-modify-attributes command, 3-29, 5-97
iaas-reboot-vservers command, 3-25, 5-100
iaas-receive-message-from-vserver command, 3-26, 5-102
iaas-release-ip-addresses command, 3-8, 5-104
iaas-run-vserver command, 3-22, 5-107
iaas-run-vservers command, 3-22, 5-110
iaas-send-messages-to-vserver command, 3-25, 5-113
iaas-start-vservers command, 3-24, 5-115
iaas-stop-vservers command, 3-24, 5-117
iaas-terminate-vservers command, 3-22, 5-119
ImportKeyPairRequest action, 3-27, 4-65
ImportVolumeRequest action, 3-16, 4-67

J

Java client API
See Cloud infrastructure Java API, 1-1

K

Key pair management, 1-4, 3-26
CLI commands
 iaas-create-key-pair, 3-26, 5-24
 iaas-delete-key-pair, 3-27, 5-45
 iaas-describe-key-pairs, 3-28, 5-64
 iaas-import-key-pair, 3-27, 5-93
Cloud infrastructure API actions
 CreateKeyPairAsObjectRequest, 4-14
 CreateKeyPairToFileRequest, 4-15
 DeleteKeyPairRequest, 3-27, 4-26
 DescribeKeyPairsRequest, 3-28, 4-43
 ImportKeyPairRequest, 3-27, 4-65
 RegisterKeyPairRequest, 3-26, 4-76
creating, 3-26
deleting, 3-27
importing, 3-27
viewing information, 3-28
Key pairs, 3-26
 create, 3-26
 delete, 3-27
 import, 3-27
 view information, 3-28

M

Modify attributes
 high availability, 3-21
ModifyAttributesRequest action, 3-29, 4-69

P

Package
 CLI, 2-2
 cloud infrastructure Java API, 2-1
Private vNets, 3-6
Public networks, 3-6

R

RebootVserversRequest action, 3-25, 4-71
ReceiveMessageFromVserverRequest action, 3-26, 4-72
RegisterAccessKeyRequest action, 3-4, 4-74
RegisterKeyPairRequest action, 3-26, 4-76
RegisterServerTemplateFromUrlRequest action, 3-11, 4-78
RegisterServerTemplateFromVserverRequest action, 3-10, 4-80
RegisterServerTemplatesFromAssemblyRequest action, 3-11, 4-82
ReleaseIpAddressesRequest action, 3-8, 4-84
RunVserverRequest action, 3-22, 4-85
RunVserverRequests action, 3-22
RunVserversRequest action, 4-88

S

SendMessageToVserverRequest action, 3-25, 4-91
Server template management
 deleting, 3-12
Server template management, 1-3, 3-9
 CLI commands
 iaas-create-server-template-from-assembly, 3-11, 5-26
 iaas-create-server-template-from-url, 3-11, 5-28
 iaas-create-server-template-from-vserver, 3-10, 5-30
 iaas-delete-server-template, 3-12, 5-47
 iaas-describe-server-templates, 3-12, 5-67
 Cloud infrastructure API actions
 DeregisterServerTemplateRequest, 3-12, 4-32
 DescribeServerTemplatesRequest, 3-12, 4-45
 RegisterServerTemplateFromUrlRequest, 3-11, 4-78
 RegisterServerTemplateFromVserverRequest, 3-10, 4-80
 RegisterServerTemplatesFromAssemblyRequest, 3-11, 4-82
 viewing information, 3-12
 Server templates, 3-1
 assembly, 3-11
 attributes, 3-12
 delete, 3-12
 management, 1-3, 3-9
 upload, 3-10, 3-11
 view information, 3-12
 Server templates management
 uploading, 3-10, 3-11
Snapshot management, 1-4, 3-13
 CLI commands
 iaas-create-snapshot, 3-17, 5-32
 iaas-delete-snapshot, 3-17, 5-49
 iaas-describe-snapshots, 3-18, 5-70
 Cloud infrastructure API actions
 CreateSnapshotRequest, 3-17, 4-16
 DeleteSnapshotRequest, 3-17, 4-27
 DescribeSnapshotsRequest, 3-18, 4-47
 creating, 3-16
 deleting, 3-17
 viewing information, 3-17
Snapshots, 3-13
 attributes, 3-17
 create, 3-16
 delete, 3-17
 management, 1-4
 view information, 3-17
StartVserversRequest action, 3-24, 4-93
Status, 3-2
StopVserversRequest action, 3-24, 4-94
Storage, 3-13

T

Tag management, 1-5
 CLI commands

- iaas-create-tags, 3-30, 5-35
- iaas-delete-tags, 3-30, 5-51
- iaas-describe-tags, 3-30, 5-72
- Cloud infrastructure API actions
 - CreateTagsRequest, 3-30, 4-18
 - DeleteTagsRequest, 4-28
 - DescribeTagsRequest, 3-30, 4-49
- creating, 3-30
- deleting, 3-30
- viewing information, 3-30
- Tags, 3-2, 3-29
 - create, 3-30
 - delete, 3-30
 - management, 1-5
 - view information, 3-30
- TerminateVserversRequest action, 3-22, 4-96

V

- vDC
 - See Virtual datacenters, 1-2
- vDC management, 1-3
 - CLI command
 - iaas-describe-vdc-capabilities, 3-5
 - CLI commands
 - iaas-describe-vdc-capabilities, 5-75
 - Cloud infrastructure API actions
 - DescribeVdcCapabilitiesRequest, 3-5, 4-51
 - viewing capabilities, 3-5
- vIP
 - See Virtual IP, 1-4
- Virtual datacenters, 1-2
 - capabilities, 3-5
 - infrastructure administration, 1-1
 - management, 1-3
 - resources utilization, 1-1
 - virtualization technologies, 1-2
- Virtual IP
 - allocate, 3-8
 - deallocate, 3-8
 - management, 1-4, 3-8
 - view information, 3-9
- Virtual IP management, 1-4, 3-8
 - allocating, 3-8
 - CLI commands
 - iaas-allocate-ip-addresses, 3-8, 5-16
 - iaas-describe-ip-addresses, 3-9, 5-62
 - iaas-release-ip-addresses, 3-8, 5-104
 - Cloud infrastructure API actions
 - AllocateIpAddressesRequest, 3-8, 4-5
 - DescribeIpAddressesRequest, 3-9, 4-41
 - ReleaseIpAddressesRequest, 3-8, 4-84
 - deallocating, 3-8
 - Viewing information, 3-9
- Virtual network management, 1-4, 3-6
 - CLI commands
 - iaas-create-vnet, 3-6, 5-37
 - iaas-delete-vnet, 3-7, 5-53
 - iaas-describe-vnets, 3-7, 5-77
 - Cloud infrastructure API actions

- CreateVnetRequest, 3-6, 4-20
- DeleteVnetRequest, 3-7, 4-30
- DescribeVnetsRequest, 3-7, 4-53
- creating, 3-6
- deletion, 3-7
- viewing information, 3-7
- Virtual networks, 3-1
 - attributes, 3-7
 - create, 3-6
 - delete, 3-7
 - management, 1-4, 3-6
 - viewing information, 3-7
- Virtual server
 - See vServers, 1-4
- Virtual storage, 3-1
- vNet
 - See Virtual networks, 1-4
- Volume management, 1-4, 3-13
 - CLI commands
 - iaas-attach-volumes-to-vserver, 3-15, 5-19
 - iaas-create-volume, 3-13, 5-40
 - iaas-delete-volume, 3-14, 5-55
 - iaas-describe-volumes, 3-14, 5-80
 - iaas-detach-volumes-from-vserver, 3-15, 5-91
 - iaas-import-volume, 3-16, 5-95
 - Cloud infrastructure API actions
 - DescribeVolumesRequest, 3-14
 - Cloud infrastructure API actions
 - AttachVolumesToVserverRequest, 3-15, 4-7
 - CreateVolumeRequest, 3-13, 4-22
 - DeleteVolumeRequest, 3-14, 4-31
 - DescribeVolumesRequest, 4-55
 - DetachVolumesFromVserverRequest, 3-15, 4-63
 - ImportVolumeRequest, 3-16, 4-67
 - creating, 3-13
 - deleting, 3-14
 - importing volumes, 3-16
 - viewing information, 3-14
- Volumes, 3-13
 - attach, 3-15
 - attributes, 3-14
 - create, 3-13, 3-16
 - delete, 3-14
 - detach, 3-15
 - import, 3-16
 - management, 1-4
 - view information, 3-14
- vServer management, 1-4
 - CLI commands
 - iaas-describe-vserver-metrics, 3-24, 5-83
 - iaas-describe-vservers, 3-23, 5-86
 - iaas-reboot-vservers, 3-25, 5-100
 - iaas-receive-message-from-vserver, 3-26, 5-102
 - iaas-run-vserver, 3-22, 5-107
 - iaas-run-vservers, 3-22, 5-110
 - iaas-send-messages-to-vserver, 3-25, 5-113
 - iaas-start-vservers, 3-24, 5-115
 - iaas-stop-vservers, 3-24, 5-117
 - iaas-terminate-vservers, 3-22, 5-119

- Cloud infrastructure API actions
 - DescribeVserverMetricsRequest, 4-57
 - DescribeVserverMetricssRequest, 3-24
 - DescribeVserversRequest, 3-23, 4-59
 - RebootVserversRequest, 3-25, 4-71
 - ReceiveMessageFromVserverRequest, 3-26, 4-72
 - RunVserverRequest, 3-22, 4-85
 - RunVserverRequests, 3-22
 - RunVserversRequest, 4-88
 - SendMessageToVserverRequest, 3-25, 4-91
 - StartVserversRequest, 3-24, 4-93
 - StopVserversRequest, 3-24, 4-94
 - TerminateVserversRequest, 3-22, 4-96
- creating, 3-21
- deleting, 3-22
- detaching volumes, 3-15
- rebooting, 3-25
- receiving messages, 3-25
- sending messages, 3-25
- starting, 3-24
- stopping, 3-24
- viewing information, 3-22
- viewing metrics, 3-23
- vServer types
 - attributes, 3-20
 - CLI commands
 - iaas-describe-vserver-types, 3-21, 5-89
 - Cloud infrastructure API actions
 - DescribeVserverTypesRequest, 3-21, 4-62
 - view information, 3-20
- vServers, 3-1
 - access, 3-26
 - access management, 1-4
 - attributes, 3-22, 3-23
 - create, 3-21
 - delete, 3-22
 - high availability, 3-21, 3-29
 - management, 1-4, 3-20
 - reboot, 3-25
 - receive message, 3-25
 - send message, 3-25
 - start, 3-24
 - stop, 3-24
 - view information, 3-22
 - view metrics, 3-23
- vServers management, 3-20
 - attaching volumes, 3-15

W

Web service

- See Cloud infrastructure API, 1-1

