Oracle® Enterprise Manager Ops Center

Cloud Infrastructure API and CLI Reference Guide

12*c* Release 1 (12.1.1.0.0)

E25150-03

June 2012

ORACLE®

Oracle Enterprise Manager Ops Center Cloud Infrastructure API and CLI Reference Guide, 12*c* Release 1 (12.1.1.0.0)

E25150-03

Copyright © 2007, 2012, Oracle and/or its affiliates. All rights reserved.

Primary Author: Karen Orozco

Contributing Author: Luis Ramos

Contributor:

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

4 Cloud Infrastructure API Reference

5   Cloud Infrastructure CLI Reference

A A Web Service Client Example

List of Examples

List of Figures

List of Tables

Preface

This guide covers installation and verb reference 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 compute, 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 following documents in the Oracle Enterprise Manager Ops Center documentation set:

- *Oracle Enterprise Manager Ops Center Release Notes*

- *Oracle Enterprise Manager Ops Center Installation Guide for Linux Operating Systems*

- *Oracle Enterprise Manager Ops Center Installation Guide for Oracle Solaris Operating Systems*

- *Oracle Enterprise Manager Ops Center Administration Guide*

- *Oracle Enterprise Manager Ops Center Concepts Guide*

- *Oracle Enterprise Manager Ops Center Feature Reference Guide*

- *Oracle Enterprise Manager Ops Center How To Library*

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.
italic	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.

1

Introduction

Oracle Enterprise Manager Ops Center provides APIs and a Command Line Interface (CLI) to enable the access to a subset of the virtual datacenter(vDC) functionality provided to cloud users.

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

This guide provides information about:

- Cloud Infrastructure API – A programatic Web service interface for managing virtual datacenter (vDC) resources allocated in vDC accounts.

- Cloud Infrastructure Client Java API – A Java client API for interacting with the Web services interface.

- Cloud Infrastructure CLI – A Java-based CLI that wraps the Cloud Infrastructure Java API with simple scripts.

The following topics are covered in this chapter:

- vDC Management Overview

- How APIs and CLI Work

- Functionality Overview

vDC Management Overview

Oracle Enterprise Manager Ops Center supports cloud management functions through its vDC Management system in three different aspects:

- 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 resources 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 related concepts used in this guide:

- Virtual Datacenter - Consolidation of virtualization servers, storage, and network resources to be utilized 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 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. 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 supports the management of the resources in vDC accounts 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
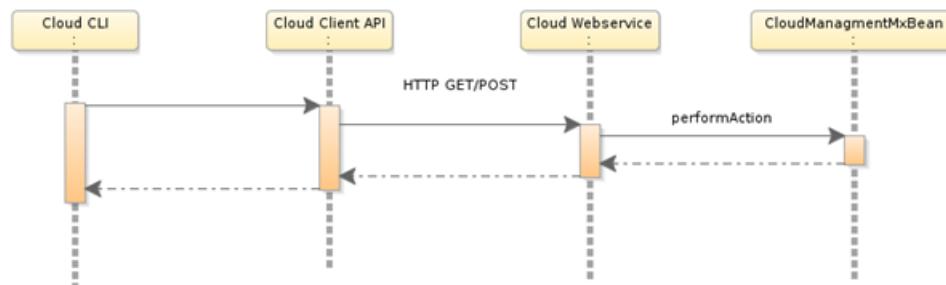
- The Cloud Infrastructure CLI is delivered as a standalone package and can be used without the need of an Oracle Enterprise Manager Ops Center installation in the computer where the CLI is deployed. This element offers the same functionality as the Cloud Infrastructure API and Java Client API.

- 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 also delivered as a standalone package.

- Cloud Infrastructure API is a Web service that exposes a subset of the functionality of the virtual datacenter management system. A Cloud Infrastructure

API request triggers an action by calling the corresponding action of the virtual center 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. On return, the caller gets at least the identifiers of the changed or created resources.

The creation of the resource IDs and the mapping to jobs are handled by the vDC management system. The Cloud Infrastructure API must wait actively for notifications from Job Manager to get the IDs of the changed or created resources. Jobs are tailored in a special way to deliver fast results, the Cloud Infrastructure API waits only until the necessary results are available.

The Cloud Infrastructure API returns a valid identifier, but the resource might have failed to be created. A cloud user must be able to view the resource information, such as name, description, and status using the identifier returned, see also Status of Account Resources.

Functionality Overview

The virtual datacenter management system in Oracle Enterprise Manager Ops Center has a set of functions that are also reached by the Cloud Infrastructure APIs and CLI. Table 1–1 to Table 1–11 show a comparison between the functions provided by the virtual datacenter system and, 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   Template Management

Functionality	Virtual Datacenter Management	Cloud Infrastructure APIs and CLI
Upload, update, and delete server template	Yes	Yes

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	Yes	Yes
Suspend and resume vServer	Yes	No
Shut down all vServers	Yes	No
Send and receive message from vServer	No	Yes

Table 1–5   Volume Management

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

2

Getting Started

This chapter identifies the prerequisites for the Cloud Infrastructure APIs and CLI. The chapter also describes how to install and configure the APIs and CLI.

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 are planned to 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 Client Java API

The Java API is delivered as a separate package as part of the Oracle Enterprise Ops Center product. This package can be installed as a standalone package in a different computer with connection to the Enterprise Controller.

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

To install this package, follow these steps as root user:

- For 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 will be stored at /opt/oracle/iaas/iaas-java-api.

Before using the API jar file set the *JAVA_HOME* environment variable and ensure that it is part of your PATH.

Cloud Infrastructure CLI

The Cloud Infrastructure CLI is delivered as a separate package as part of the Oracle Enterprise Ops Center product. This package can be installed as standalone package in a different computer with connection to the Enterprise Controller.

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

To install this package, follow these steps as root user:

- For 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 will be stored at /opt/oracle/iaas/cli. The user in charge of executing the CLI commands needs to have permissions to access this directory.

Before using the Cloud Infrastructure CLI:

1. Set *JAVA_HOME* environment variable and ensure that it 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 *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 using the APIs or CLI, the following tasks must be completed by your cloud administrator:

1. Create vDC and an account for the vDC.

2. Create a cloud user an associate it 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, see "Creating an Access Key".

3

Managing Resources

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

For a complete reference and other examples using 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 Client Java API you can refer to the javadoc files included in the package of the API.

Resources Overview

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

- Access Keys – Used to manage a 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 – Provide pre-built images for creating vServers. 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) – An entity that 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 also have the following information:

- Attributes – When a resource is created, the resource has a set attributes that describes it, includes defined entities such as name, description, and size. Some of these attribute 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 may 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 creating 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 when an account resource is usable or in unusable state.

The are different states an account resource can have:

Pending Status

Resource creation is in progress.

■ SCHEDULED – Indicates that the creation job is in progress. Wait for the resource transition from this state before using.

■ FAILED – Indicates that the creation of the resource failed, a failed job is reflected in the BUI.

Healthy Status

A resource exists and the following values can occur to indicate a resource is usable:

■ OK – Indicates that the resources is healthy and usable as normal.

■ INFO – Indicates that the resource is healthy but an informational event is awaiting attention in the BUI.

> **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 and the following values can occur to indicate a resource might not be usable. This is not applicable for WARNING state.

> **Note:** The particular status that are applicable are dependent 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, terminating a vserver can exhibit a transition to the vserver run state SHUTDOWNDETACHED before the resource is removed.

A failed delete operation could also fail in which case the resource must transition to one of the unhealthy states above.

Managing Account Access

User management is provided by a central service of Oracle Enterprise Manager Ops Center. The central entry point for APIs or CLI requests is the Web service. The Web service is responsible for the authentication and authorization of a cloud user request by using access keys. A cloud user must create an access key for an account to perform any other action for the account resources allocated in that account.

An access key comprises by 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 allowing user name and password authentication.

The APIs and CLI provides 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 perform the operations listed above 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 for authenticating 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 using the CreateAccessKey method in the CLI, after the access key creation, the private key is provided to the user inside a KeyStore for future use.

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

Whichever access key importation 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 action and the Cloud Infrastructure CLI command required to create an access key are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RegisterAccessKeyRequest	akm-create-access-key
CreateAccessKeyAsObjectRequest	

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

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=RegisterA
ccessKey&Version=1&Timestamp=1330975344&Expires=1333975344&account=ACC-0162da5a-5d
25-4096-af59-3dd1de27cfad&publicKey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuf
Vdjdp0MmOLbNypLVMWXfmhusawid4Wg4n4FZewSmoBEYA8f8wIA0SI87Shi7RtMcWsEoXvNNHA0wcJoA1R
jyVLsI3rtrq0c0k7AxQSwb4UK/rSXW1NXxMh/mE7b3gdA6d9VuwIPnZJ5ZFQUZCLyhaAotLCdACrzbgzYX
dqt+rstutT1AVkE2UAMcm5O3KnIoObZKb8JtepSt74A9RgVBkcCBjmKGfLNOL1KlZconkITm85TWKRaGRF
uASxdl2ZrD723ZNb66X/a9ebxTMr6vVeskcaZpPlHzvgMOpiyDGwRvxn9yM5WB83zFDGT26Lihn/bKzLJX
a+F2YNkLrTJQIDAQAB
```

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

```
akm-create-access-key  --base-url https://<EnterpriseControllerHostname>/ --user
clouuser1 --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 of the access keys the user owns.

When viewing access key information, a list of the access keys found is displayed 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	akm-describe-access-keys

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. When you delete an access key, the registered public key is deleted and invalid. 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	akm-delete-access-key

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

```
https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=DeleteAcc
essKey&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 is allowed to view details about the capabilities of the vDC for an account. The vDC capabilities listed include:

- Virtualization Type
- Virtualization Version
- Processor Architecture
- Processor Version
- Distribution Group Support

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	iaas-describe-vdc-capabilities

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVdcCapabilities&Versio
n=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 vNets – A private vNet is created based on the private network from the network domain of the 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 networks – This type of networks can be shared among a number of accounts in a vDC. vServers members of public networks have also external communication beyond vDCs and can be used to host public services.

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

APIs and CLI provides operations for:

- Creating a vNet

- Deleting a vNet

- Viewing vNets 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	iaas-delete-vnet

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 vNets 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 vNets found 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	iaas-describe-vnets

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. The vIP can be either from a public network or from a private vNet.

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 done 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.

APIs and CLI provides 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, the allocated vIP address can be statically assigned to a 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&AccessKeyI
d=AK_
32&vnet=VNET-6ea466f5-6e6b-4159-adf3-8867473d4cf4&Version=1&Timestamp=132034220680
8&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&SignatureM
ethod=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 non allocated IP addresses associated with vServers, use the actions for 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 addresses: 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	iaas-describe-ip-addresses

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeIpAddresses&AccessKeyI
d=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 will be installed when creating the vServer. Server templates are specific to processor architecture of the server pool and virtualization type and may be pre-built images or identify the OS 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 and 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 based on:

- An existing vServer.

- From a single virtual machine template.

- From an assembly. When uploading an assembly, snapshots can be also created as result.

APIs and CLI provides 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, installs some additional software, or configures certain aspects, the user stops the vServer to get a target state, that state must be conserved and reusable. 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	iaas-create-server-template-from-vserver

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplateFromVser
ver&AccessKeyId=AK_
2&name=mySTfromVserver&vserverId=VSRV-fdba0b48-6e1c-4f41-bde4-2c739dfeeb2&Version=
1&Timestamp=1324422880552&Expires=1324423180552f&SignatureMethod=SHA512withRSA&Sig
natureVersion=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 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	iaas-create-server-template-from-url

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%2
Fca-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 inter connectivity. An assembly is contained in a single .ova (Open Virtualization Format Archive) file. The assembly file must be accessible through a URL.

When uploading 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 will have a tag set to indicate which assembly it belongs to. This tag name is assemblyKey, and it's value is the original assembly ID returned for this action.

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
RegisterServerTemplatesFromAssemblyRequest	iaas-create-server-template-from-assembly

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterServerTemplatesFromAss
embly&AccessKeyId=AK_
3&name=myST2&url=http%3A%2F%2Fadc4120293.us.oracle.com%3A8888%2FmyAssembly.ova&Ver
sion=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	iaas-delete-server-template

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeregisterServerTemplate&Acces
sKeyId=AK_
3&serverTemplateId=TMPL-f089b985-f7fc-4b8a-a5f8-df8f44c95f3c&Version=1&Timestamp=1
320098301701&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-eeeeeeeeee03
```

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 templates 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	iaas-describe-server-templates

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeServerTemplates&Versio
n=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 and they are immutable.

APIs and CLI provides 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. cloud users can also create a shared or non-shared 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&Timesta
mp=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 deletion of a volume does not affect any snapshot that has been created previously based on that volume. The snapshots 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-create-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

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

You can attach volumes to a vServer during the vServer creation process. If a vServer is created before volume creation, then a cloud user can attach a volume to a stopped vServer. After attaching, the volume is available as a device in the vServer.

The Cloud Infrastructure API action and the Cloud Infrastructure CLI command required to attach a volume to a stopped vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
AttachVolumesToVserverRequest	iaas-attach-volumes-to-vserver

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-052cb4b4-5e56-430
3-8b3a-82d6ba743a15&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&AccessKeyI
d=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 stopped vServer. After detaching, the volume is no longer available as a device in the vServer. 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	iaas-detach-volumes-from-vserver

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=DetachVolumesFromVserver&Versi
on=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

Snapshots availability and management is only available if supported by the storage of the vDC.

 Taking 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 as result of uploading an assembly.

In terms of snapshots created from an assembly, there are four types of exposed disks in an assembly's template. Three of them 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 - Extra disk with already present data.

- Private Raw - Dynamically created as empty disk, no disk data, just a size.

- Shared Raw - Dynamically created as empty disk, no disk data, just a size.

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 nor any other snapshots of the same volume. 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 snapshot 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&Ti
mestamp=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 within a distribution group are placed on separate server of the server pool the vDC uses. If a spare server is not available, then the create vServer job fails.

APIs and CLI provides operations for:

- Creating a Distribution Group
- Deleting a Distribution Group
- Viewing Distribution Groups 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.

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&Versio
n=1&Timestamp=1324407098313&Expires=1324407398313&name=myDistributionGroup&AccessK
eyId=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	iaas-delete-distribution-group

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=DeleteDistributionGroup&Versio
n=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 Groups 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

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	iaas-describe-distribution-groups

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeDistributionGroups&Ver
sion=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 full 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 a temporary local storage. For hard disk storage, the associated volume is used. A vServer can have one or more volumes attached.

APIs and CLI provides operations for:

- Viewing vServer Types
- Creating a vServer

- [Deleting a vServer](#)
- [Viewing vServer Information](#)
- [Stopping a vServer](#)
- [Starting a vServer](#)
- [Rebooting a vServer](#)
- [Sending Messages to a vServer](#)
- [Receiving 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.

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	iaas-describe-vserver-types

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 or multiple vServer at once based on a server template. When a vServer is created, it is automatically started.

When creating a single vServer an allocated vIP address must be used. When creating multiple vServer an IP address is automatically assigned from each specified vNets.

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

- A vServer type

- A server template

- One or more vNets

- A key pair

There are other resources that a cloud user can specify when creating a vServer that depends on the needs of the user. This other resources are:

- One or more volumes

- A Distribution group

The Cloud Infrastructure API action and the Cloud Infrastructure CLI command required to run a vServer are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
RunVserverRequest or RunVserversRequest	iaas-run-vserver or iaas-run-vservers

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=VNE
T-84ada392-1c13-4f86-8365-1cf7f9c8aadf&name=myVserver&ipAddresses.1=192.168.0.2&se
rverTemplateId=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-vservers --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	iaas-terminate-vservers

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

```
https://<EnterpriseControllerHostname>/iaas/?Action=TerminateVservers&Version=1&Ti
mestamp=1321385662260&Expires=1321385962260&vserverIds.1=VSRV-0fb57293-347c-4717-9
6ef-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 type are:

- ID
- Name
- Description
- Status
- vNets
- IP Addresses
- Server Template ID
- Key Name
- vServer Type

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&Tim
estamp=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
```

Stopping a vServer

A cloud user can stop a vServer. When a vServer is stopped, the guest operating system is shutdown. A stopped vServer can be restarted at a later point. 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 3–63   Stopping a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=StopVservers&Version=1&Timesta
mp=1320105610783&Expires=1320105910783&vserverIds.1=VSRV-c1e236e6-ef4d-4936-911a-9
7923dfbc291&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–64   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 or restart 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 3–65   Starting a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=StartVservers&Version=1&Timest
amp=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 3–66   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 3–67   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 3–68   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 Messages to a vServer

A cloud user can send a message to a vServer, this messages can be read from the guest OS. The message is sent to a common buffer in the virtualization layer that have a limited size. This action might return an error if the size limits 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
SendMessagesToVserverRequest	iaas-send-messages-to-vserver

Example 3–69   Sending messages to a vServer using the Cloud Infrastructure API

```
ttps://<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
```

Example 3–70   Sending messages to a vServer using the Cloud Infrastructure CLI

```
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
```

Receiving 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 3–71   Receiving message from a vServer using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=ReceiveMessageFromVserverReque
st&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 3–72   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. By using key pair management, a way to get the credentials for user authentication into the guest operating system is defined. Key pairs are used to restrict access to vServers started up 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. The key pairs are managed per account and are only visible as tags.

APIs and CLI provides 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 RSA 2048 bit key. If a different key pair strengths are 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 3–73   Creating a key pair using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=RegisterKeyPair&Version=1&Time
stamp=1318279380444&Expires=1318279680444&publicKey=ssh-rsa+AAAAB3NzaC1yc2EAAAADAQ
ABAAAABAQDg2eienGE4vEMSMcVMbYbC8z2q%2Fvhz3H6AanlJ6B4udseK%0A8CpaHJ23eGWcjcgAmuZCJ%2
FOoHUA2dN2PNPuK6g%2BZndR8wVaaQT89eWDZx9oaf0%2F2Eg%2FLeKJ3moVH%0AvIYvFB9aFCpa4H%2BO
mLfM%2FmfQ4CYeDfo0r0jxCCB0YLo0876LQqK5X%2BtgRXwbAbPH2Mzbzp%2FzzdkQ%0ArsBqSgUQ%2B1V
4LkN6TQe06P5a2QYIlUhRXwUorTnbXczGq9zEJJ7ef%2F74xIQZfAipkYkyGgktsXrM%0A%2F%2Bs789v9
ipaDB5B26y3aqjIdvW4ZLDvuGXPs60aiUfj2WGIqx0KSVL%2FyB%2FtK1WbuZYwl+IaaS-Generated&ke
yName=myKeyPairK&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–74   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.

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
IImportKeyPairRequest	iaas-import-key-pair

Example 3–75   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
AAAQEAtJnFD8INGLtM%2FQIlxkDjh4t2R4%2FtTMpUDzMIRP%0A&keyName=myKeyPair&AccessKeyId=
AK_1&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_ READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–76   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. By deleting a key pair 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 3–77   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 3–78   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 of a key pair are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeKeyPairsRequest	iaas-describe-key-pairs

Example 3–79   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 3–80   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.

APIs and CLI provides 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. For viewing 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 a vNet are:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeAttributesRequest	iaas-describe-attributes

Example 3–81   Viewing attributes of an account resource using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeAttributes&Version=1&T
imestamp=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
```

Example 3–82   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.

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 3–83   Modifying vNet attributes using the Cloud Infrastructure API

```
https://<EnterpriseControllerHostname>/iaas/?Action=ModifyAttributes&Version=1&Tim
estamp=1318464546890&Expires=1318464846890&resourceId=VOL-246b5c62-4072-41cf-885b-
99d6c63583bd&attributes.2.value=myNewDescription&attributes.2.name=description&att
ributes.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 3–84   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 --attr-names 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 in a resource using their tag names or tag values. Tags are used to bind user specific information to entities.

APIs and CLI provides 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 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	iaas-create-tags

Example 3–85   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-ee3d3
98e0d50&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 3–86   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 is:

Cloud Infrastructure API Action	Cloud Infrastructure CLI Command
DescribeTagsRequest	iaas-delete-tags

Example 3–87   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-ee3d3
98e0d50&tags.1.name=myTag&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_
FOR_THE_ READABILITY_OF_THE_
REQUEST&SignatureMethod=SHA512withRSA&SignatureVersion=1
```

Example 3–88 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	iaas-describe-tags

Example 3–89 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 3–90 Viewing tag information using the Cloud Infrastructure CLI

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

4

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.

Web Service Overview

The Cloud Infrastructure API is a Web service API implemented in Oracle Enterprise Manager Ops Center that exposes its vDC Management System service model. This Web service provides access to a subset of the vDC Management system functionality that is available through the Oracle Enterprise Manager Ops Center UI.

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 in Ops Center requires an SSL encrypted HTTP connection. A typical request to the Cloud Web Service is made through HTTP using a GET or POST method including the request parameters for the requested 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 two servlets (interfaces) deployed into the Common Agent Container of Ops Center (cacao):

- The Access Key Management (AKM) servlet, this servlet handles all user name and password authentication action.

- The IAAS servlet, this servlet handles all cloud computing related actions.

Required Web Service HTTP Request Parameters

The common parameters that must be included in a HTTP request to the Web Service are:

Servlet	Parameter	Description
AKM and IAAS	Action	Indicates the name of the IAAS or AKM action to perform
IAAS	Version	Version of the Web Service.

Servlet	Parameter	Description
IAAS	AccessKeyId	Access Key ID. This identifies the user and the account. This value returned by the akm RegisterAccessKey action.
AKM	Timestamp	Timestamp in milliseconds since 1st of January 1970.
AKM	Expires	Time at which the request expires. Expiry date in milliseconds since 1st of January 1970, must be greater than the Timestamp.
IAAS	Signature	Encoded Hash value of the data to be signed. Signature is used to authenticate the data.
IAAS	SignatureMethod	Type of the signature.
IAAS	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. 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. URL encode all parameter names and their values. The following characters are not URL encoded: A-Z, a-z, 0-9, minus '-', underscore '_', period '.', and asterisk '*' and remain as they are. All other characters are encoded as %XX (with XX being a hexadecimal number consisting of 0-9 and/or uppercase A-F). Extended UTF-8 characters are encoded with %XX%XX. Space is encoded as '+'. 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. "POST" or "GET", depending on request type. "\n"

   b. Hostname as given in the HTTP host header field, all in lower case (for example, `opscenter.foo.bar`) "\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. The signature method must be "SHA512withRSA".

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

Examples of Web Service Requests

Example 4–1   URL for a 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/XRQ9LxyYeBgzjO3lHP+hFhUo+gUtQaS
YPhUHH7eTkxg/CrolMxibglypJM/rIf90yEqSeqhphQt7hWxlT0DNAy6/cZt8isT/Tu8V7ZFjBFkEpLfN9
```

```
7bIOJ2vIIpOeetmftuw4ObtqjbUp6+7dpVkhhCQnX0MAIDj+mjorEOzcwK+F1pYuzES0fjaW0MowG+cA/9
gttDjg7r5H29i3qbbjIlvAt6fk1HPpSxQTSTOTg==
```

Example 4–2   URL for a HTTP GET request to the AKM servlet:

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

For a detailed example for generating requests to the Web Service, see A Web Service Client Example.

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, SendMessagesToVserverRequest, 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 addresses from a vNET.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=AllocateIpAddresses&Version=1&
Timestamp=1320342206808&Expires=1320342506808&vnet=VNET-6ea466f5-6e6b-4159-adf3-88
67473d4cf4&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</ipAddres
ses></result>
```

See Also

- DeleteVnetRequest
- DescribeIpAddressesRequest
- DescribeVnetsRequest
- ReleaseIpAddressesRequest

AttachVolumesToVserverRequest

Description

Action to attach one or more volumes to a stopped 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 attachs a volume to the specified vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=AttachVolumesToVserver&Version
=1&Timestamp=1321384886152&Expires=1321385186152&volumeIds.1=VOL-052cb4b4-5e56-430
3-8b3a-82d6ba743a15&vserverId=VSRV-8ae29df9-ccfe-4184-acb8-10080665d7f6&AccessKeyI
d=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. User using this option must have 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 provided for this action, refer to RegisterAccessKeyRequest action.

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
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 provided for this action, refer to RegisterAccessKeyRequest action.

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*

Default: *None*

Required: *Yes*

description
Description of the distribution group.

Type: *GenericString*

Default: *None*

Required: *No*

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&Versio
n=1&Timestamp=1324407098313&Expires=1324407398313&name=myDistributionGroup&AccessK
eyId=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>
```

See Also

- DeleteDistributionGroupRequest
- DescribeDistributionGroupsRequest

CreateKeyPairAsObjectRequest

Description

Action to create a key pair. This is the only action which is client side only and will not travel the network.

Request Parameters

keyName
Name of the key pair to create.

Type: *xs:strin*g

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 provided for this action, refer to RegisterKeyPairRequest action.

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 provided for this action, refer to RegisterKeyPair Request action.

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*

Default: *None*

Required: *Yes*

description
Description of the snapshot

Type: *GenericString*

Default: *None*

Required: *No*

volumeId
ID of the volume.

Type: *VolumeIdType*

Default: None

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 name and value.

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-ee3d3
98e0d50&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
Description of the virtual network.

Type: *GenericString*

Default: *None*

Required: *No*

size
Size of the subnet of the virtual network.

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 adding a description.

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 an empty 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 GB. This option should not be used if a snapshot ID is given. An error is generated if both parameters, size and snapshotid, 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 (not shared)

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 parameters, size and snapshotid, 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*

volumeId
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&Timesta
mp=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></res
ult>
```

Example 2
The following example creates a volume base on the specified snapshot ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVolume&Version=1&Timesta
mp=1320100009343&Expires=1320100309343&snapshotId=SNAP-7a717e39-fe67-4573-a93d-889
b3446176b&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></res
ult>
```

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 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&Versio
n=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&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
```

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-ee3d3
98e0d50&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-b3f4-195a090bbc0
1&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&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
```

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&Versi
on=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&Ti
mestamp=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 or, account attribute name and value.

Request Parameters

account
List of account IDs.

Type: *AccountIdType*

Default: *None*

Required: *No*

forUser
User name of a different cloud user. 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 examples 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 examples 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 one or more of attributes of a resource.

Note: This action is restrained to following resources 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&T
imestamp=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 attributes of existing distribution groups for an account. This action allows filtering of distribution groups displayed by ID or by attribute names and values.

Request Parameters

ids
List of distribution groups.

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: *DescribeKeyPairsResultItem*

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&Ver
sion=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></items></result>
```

See Also

- CreateDistributionGroupRequest

- DeleteDistributionGroupRequest

DescribeIpAddressesRequest

Description

Action to display reserved 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 the reserved IP addresses and their attributes

Type: *DescribeIpAddressesResultItem*

Minimum: *0*

Maximum: *Unbounded*

Examples

Example 1
The following example lists all reserved 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-ffffffffff02</vnet></items><items><ipAddress>10.1.0.1</i
pAddress><vnet>VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff02</vnet></items><items><ipA
ddress>10.1.0.0</ipAddress><vnet>VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff02</vnet><
```

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

Example 2

The following example displays the reserved 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-415
9-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&Tim
estamp=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&Versio
n=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>Sen
dReceiveImage.tar.gz</name><status>OK</status><size>-1</size><public>false</public
><imageType>VMTemplate</imageType><readOnly>false</readOnly></items><items><id>TMP
L-48cfe488-b93d-4085-aa2e-0a3b39d511cd</id><name>OVM_EL52_jeos_i386_PVM_
WebLogic10gR3_v10.tar.gz</name><description>Import URLs:
[http://gridware.germany.sun.com/vm-templates/WebLogic/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&Versio
n=1&Timestamp=1320088942728&Expires=1320089242728&ids.1=TMPL-e7df3178-12ea-48cc-8a
1a-6b3f8256dd0f&AccessKeyId=AK_3&Signature=SIGNATURE_HAS_BEEN_REMOVED_FOR_THE_
READABILITY_OF_THE_REQUESTSignatureMethod=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>Sen
dReceiveImage.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 snapshots 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 all available snapshots for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeSnapshots&Version=1&Ti
mestamp=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 tag attributes of the resources with a tag associated with them. 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 virtual snapshot attributes 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&Timesta
mp=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</reso
urceId><name>oc.internal.oracle.cloud.security.group.hidden</name><value>private</
value></items><items><resourceId>VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50</resour
ceId><name>myTag</name><value>myTagValue</value></items><items><resourceId>VNET-2b
8b6c91-065c-4645-9cdb-edb3afd92524</resourceId><name>oc.internal.oracle.cloud.secu
rity.group.hidden</name><value>private</value></items><items><resourceId>VNET-07e7
4ad3-1ab1-4188-915d-cbf6242a1eeb</resourceId><name>oc.internal.oracle.cloud.securi
ty.group.hidden</name><value>private</value></items></result>
```

Example 2

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

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeTags&Version=1&Timesta
mp=1320102284357&Expires=1320102584357&ids.1=VNET-9ba968d5-e64f-4b54-a639-ee3d398e
0d50&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</reso
urceId><name>oc.internal.oracle.cloud.security.group.hidden</name><value>private</
value></items><items><resourceId>VNET-9ba968d5-e64f-4b54-a639-ee3d398e0d50</resour
ceId><name>myTag</name><value>myTagValue</value></items></result>
```

See Also

- CreateTagsRequest
- DeleteTagsRequest

DescribeVdcCapabilitiesRequest

Description

Action to display the vDC capabilities for an account.

Request Parameters

filters

List of virtual data center attributes 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: *VdcCapabilitiesItem*

Minimum: *1*

Maximum: *Unbounded*

Examples

Example 1

The following example lists the vDC capabilities for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVdcCapabilities&Versio
n=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><i
tems><name>VirtualizationVersion</name><value>3.0.2</value></items><items><name>Pr
ocessorArch</name><value>Default_Intel_F6_
M23</value></items><items><name>ProcessorVersion</name><value>Default_Intel_
Family:6_Model:23</value></items></result>
```

Example 2

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

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVdcCapabilities&Versio
n=1&Timestamp=1320085633535&Expires=1320085933535&filters.1.filterName=Virtualizat
ionType&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&Acces
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 of 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></resul>
```

See Also

- CreateVnetRequest
- DeleteVnetRequest
- DescribeAttributesRequest
- ModifyAttributesRequest

DescribeVolumesRequest

Description

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

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></items><items><id>VOL-3bbc8f1c-bb4c-4a7
0-a370-4b6c06d990dc</id><name>myVol</name><status>OK</status><size>0</size></items
></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></items></result>
```

See Also

- AttachVolumesToVserverRequest

- CreateVolumeRequest

- DeleteVolumeRequest

- DescribeAttributesRequest

- DetachVolumesFromVserverRequest

- ImportVolumeRequest

- ModifyAttributesRequest

DescribeVserversRequest

Description

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

Request Parameters

ids
List of one or more vServers 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&Tim
estamp=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-062c548b-7346-4318-a802-a223288747e0</id><name>gue
stSendRecRGB_1</name><description>Oracle VM Virtual
Machine</description><status>OK</status><vnets>VNET-2b8b6c91-065c-4645-9cdb-edb3af
d92524</vnets><vserverType>OVM</vserverType></items><items><id>VSRV-ae897da4-f13d-
4a92-8464-394e0e396bf9</id><name>guestSendRec_RGB2_1</name><description>Oracle VM
Virtual
Machine</description><status>OK</status><vnets>VNET-2b8b6c91-065c-4645-9cdb-edb3af
d92524</vnets><vserverType>OVM</vserverType></items><items><id>VSRV-c1e236e6-ef4d-
4936-911a-97923dfbc291</id><name>guestVserverRGB_1</name><description>Oracle VM
Virtual
Machine</description><status>OK</status><vnets>VNET-07e74ad3-1ab1-4188-915d-cbf624
2a1eeb</vnets><vserverType>OVM</vserverType></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-062c548b-7346-4318-a802-a223
288747e0&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-062c548b-7346-4318-a802-a223288747e0</id><name>gue
stSendRecRGB_1</name><description>Oracle VM Virtual
Machine</description><status>OK</status><vnets>VNET-2b8b6c91-065c-4645-9cdb-edb3af
d92524</vnets><vserverType>OVM</vserverType></items></result>
```

See Also

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

DescribeVserverTypesRequest

Description

Action to display the permitted vServer types for an account.

Request Parameters

There is not extra parameters for this action's request.

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 a volume from a stopped vServer.

Request Parameters

vserverId
ID of the vServer.

Type: *VserverIdType*

Default: *None*

Required: *Yes*

volumeIds
List of one of more volume IDs.

Type: *VolumeIdType*

Default: *None*

Required: *Yes*

force
Option to force detach.

Type: *xs:boolean*

Default: *0*

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&Versi
on=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
```

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.

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+AAAAB3NzaC1yc2EAAAABIw
AAAQEAtJnFD8INGLtM%2FQIlxkDjh4t2R4%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:1
a: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: *boolean*

Default: *0*

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&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
```

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></res
ult>
```

See Also

- [AttachVolumesToVserverRequest](#)

- [CreateVolumeRequest](#)

- [DeleteVolumeRequest](#)

- [DescribeVolumesRequest](#)

- [DescribeAttributesRequest](#)

- [DetachVolumesFromVserverRequest](#)

- [ModifyAttributesRequest](#)

ModifyAttributesRequest

Description

Action to modify one or more attribute values of 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 of the specified volume ID.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=ModifyAttributes&Version=1&Tim
estamp=1318464546890&Expires=1318464846890&resourceId=VOL-246b5c62-4072-41cf-885b-
99d6c63583bd&attributes.2.value=myNewDescription&attributes.2.name=description&att
ributes.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 restart 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 restarts 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

Note: 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 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=ReceiveMessageFromVserverReque
st&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
- SendMessagesToVserverRequest
- StartVserversRequest
- StopVserversRequest
- TerminateVserversRequest

RegisterAccessKeyRequest

Description

Action to register an access key for a cloud user for the specified account. This action also allows the creation of an access key of another cloud user, in that case, the user running this action must have 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 examples registers an access key for an account.

HTTP Request

```
https://<EnterpriseControllerHostname>/akm/?
Action=RegisterAccessKey&Version=1&Timestamp=1317055640451&Expires=1317055700451&p
ub
licKey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3XtkY0aBtOSFZcmk
%2FPjiMj7xyGCzRB2I%0AwQXV
%2BQiY5La1Ppj8fSxrs1t85Hy2%2FUY8gfVYy3peGm1mkdo0xtFfP90ACxKAH7Z8%2B8LqgPT6w
DxY%0AxlkOF9k80M9fHxXrfWNfxfw87yd%2FNZdeZOvgRxW8B
```

%2BsuHmAV5HyRRFKuiQFko7EYVAijFIhRv7ez%0AThKNmP
%2BSrOGvPuUoG035TDloSbdpQ08ZGurvIdGevcJpM3IfOEy5tqyfU%2FdvtaLh%2FyhabUjCZN6E
%0AqGs2t9C75D9sgVy%2FykbDiXYOorRV3wUoatB43YDPpx5TjpJZIogeF1vCVETDGg6sKsvl
%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
ABAAAABAQDg2eienGE4vEMSMcVMbYbC8z2q%2Fvhz3H6AanlJ6B4udseK%0A8CpaHJ23eGWcjcgAmuZCJ%2
FOoHUA2dN2PNPuK6g%2BZndR8wVaaQT89eWDZx9oaf0%2F2Eg%2FLeKJ3moVH%0AvIYvFB9aFCpa4H%2BO
mLfM%2FmfQ4CYeDfo0r0jxCCB0YLo0876LQqK5X%2BtgRXwbAbPH2Mzbzp%2FzzdkQ%0ArsBqSgUQ%2B1V
4LkN6TQe06P5a2QYIlUhRXwUorTnbXczGq9zEJJ7ef%2F74xIQZfAipkYkyGgktsXrM%0A%2F%2Bs789v9
ipaDB5B26y3aqjIdvW4ZLDvuGXPs60aiUfj2WGIqx0KSVL%2FyB%2FtK1WbuZYwl+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</k
eyFingerprint></result>
```

See Also

- CreateKeyPairAsObjectRequest
- CreateKeyPairToFileRequest
- DeleteKeyPairRequest
- DescribeKeyPairsRequest
- ImportKeyPairRequest

RegisterServerTemplateFromUrlRequest

Description

Action to register a server template from aURL.

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 creates 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%2
Fca-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
```

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</serve
rTemplateId></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=RegisterServerTemplateFromVser
ver&Version=1&Timestamp=1324422880552&Expires=1324423180552&name=mySTfromVserver&v
serverId=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="RegisterServerTemplateFromUrlResult"
requestId="258"><serverTemplateId>TMPL-f047b985-f7fc-4b8a-a5f8-df8f44c95f5a</serve
rTemplateId></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=RegisterServerTemplatesFromAss
embly&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 reserved IP addresses from a virtual network.

Request Parameters

vnet
ID of the virtual network.

Type: *VnetIdType*

Required: *Yes*

ipAddresses
One or more reserved 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&T
imestamp=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 the server template.

Type: *ServerTemplateIdType*

Required: *Yes*

ipAddresses
List of one or more reserved IP addresses.

Type: *GenericString*

Required: *Yes*

keyName
Name of the key pair.

Type: *GenericString*

Required: *Yes*

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*

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=VNE
T-84ada392-1c13-4f86-8365-1cf7f9c8aadf&name=myVserver&ipAddresses.1=192.168.0.2&se
rverTemplateId=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

- RunVserverRequest

- SendMessagesToVserverRequest

- StartVserversRequest

- StopVserversRequest

- TerminateVserversRequest

RunVserversRequest

Description

Action to create and start multiple vServer.

Request Parameters

name
Name of the vServer.

Type: *GenericString*

Required: *Yes*

description
Description of the vServer.

Type: *GenericString*

Required: *No*

serverTemplateId
ID of the server template.

Type: *ServerTemplateIdType*

Required: *Yes*

num
Number of the 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: *Yes*

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*

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&Timestam
p=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_
REQUESTSignatureMethod=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 OS.

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&Timest
amp=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
- SendMessagesToVserverRequest
- 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*

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&Timesta
mp=1320105610783&Expires=1320105910783&vserverIds.1=VSRV-c1e236e6-ef4d-4936-911a-9
7923dfbc291&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
- SendMessagesToVserverRequest
- StopVserversRequest
- TerminateVserversRequest

TerminateVserversRequest

Description

Action to stop and 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*

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 terminates a vServer.

HTTP Request

```
https://<EnterpriseControllerHostname>/iaas/?Action=TerminateVservers&Version=1&Ti
mestamp=1321385662260&Expires=1321385962260&vserverIds.1=VSRV-0fb57293-347c-4717-9
6ef-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

- SendMessagesToVserverRequest

  StartVserversRequest

- StopVserversRequest

Data Types

The following data types are described in this section:

- AccountIdType
- DescribeAccessKeysResultItem
- DescribeAccountsResultItem
- DescribeIpAddressesResultItem
- DescribeKeyPairsResultItem
- DescribeServerTemplatesResultItem
- DescribeSnapshotsResultItem
- DescribeTagsResultItem
- DescribeVdcCapabilitiesResultItem
- DescribeVnetsResultItem
- DescribeVolumesResultItem
- DescribeVserversResultItem
- DescribeVserverTypesResultItem
- 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}

Ancestor(s)

- 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*

Ancestor(s)

- 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 none or more distribution group tags.

Type: *TagType*

Ancestor(s)

- None

Actions

- DeleteDistributionGroupRequest

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*

Ancestor(s)

- 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*

Ancestor(s)

- 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*

Ancestor(s)

- 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*

Ancestor(s)

- 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*

Ancestor(s)

■ None

Actions

■ DescribeTagsRequest

DescribeVdcCapabilitiesResultItem

Data Type Elements

name
Name of vDC capability.

Type: *GenericString*

value
Value of the vDC capability.

Type: *GenericString*

Ancestor(s)

- 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*

Ancestor(s)

- 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*

Ancestor(s)

- None

Actions

- DescribeVolumesRequest

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 none or more tags for the vServer.

Type: *TagType*

vnets
List of virtual network IDs associated with the vServer.

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*

Ancestor(s)

- 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*

Ancestor(s)

- 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}

Ancestor(s)

- 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*

Ancestor(s)

- 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*

Ancestor(s)

- FilterItem
- Message
- TagType

Actions

- CreateAccessKeyAsObjectRequest
- CreateAccessKeyToFileRequest
- CreateKeyPairAsObjectRequest
- CreateKeyPairToFileRequest
- DescribeAccessKeysRequest
- DescribeAccountsRequest
- DescribeKeyPairsRequest
- ImportKeyPairRequest
- RegisterAccessKeyRequest
- RegisterKeyPairRequest

Message

Data Type Elements

key
Key associated with the message.

Type: *GenericString*

value
The value of the message.

Type: *GenericString*

Ancestor(s)

- None

Actions

- SendMessagesToVserverRequest

PositiveInteger

Data Type Elements

This data type defines the pattern value of a PositiveInteger data type.

Type: *xs:integer*

minInclusive value: *1*

Ancestor(s)

- None

Actions

- AllocateIpAddressesRequest
- CreateVnetRequest
- CreateVolumeRequest
- RunVserversRequest

ResourceAttribute

Data Type Elements

name
Name of an attribute for specified resource.

Type: *xs:string*

value
Value for the specified resource's attribute.

Type: *xs:string*

Ancestor(s)

- 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}

Ancestor(s)

- 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}

Ancestor(s)

- 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}

Ancestor(s)

- 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*

Ancestor(s)

- DescribeServerTemplatesResultItem
- DescribeSnapshotsRequest
- DescribeVnetsResultItem
- DescribeVolumesResultItem
- DescribeVserversResultItem
- 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*

Ancestor(s)

- 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}

Ancestor(s)

- DescribeIpAddressesResultItem
- DescribeVserversResultItem

Actions

- AllocateIpAddressesRequest
- DeleteVnetRequest
- DescribeIpAddressesRequest
- DescribeVnetsRequest
- DescribeVserversRequest
- ReleaseIpAddressesRequest
- RunVserverRequest
- RunVserversRequest

VolumeIdType

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}

Ancestor(s)

- 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}

Ancestor(s)

- DescribeIpAddressesRequest

Actions

- AttachVolumesToVserverRequest
- DescribeIpAddressesRequest
- DescribeVserversRequest
- DetachVolumesFromVserverRequest
- RebootVserversRequest
- RegisterServerTemplateFromVserverRequest
- SendMessagesToVserverRequest
- StartVserversRequest
- StopVserversRequest
- ReceiveMessageFromVserverRequest
- TerminateVserversRequest

5

Cloud Infrastructure CLI Reference

This chapter can be used as reference to use the Cloud Infrastructure CLI once installed. It includes a description, syntax, options, and examples for each command.

Common Command Options

All the Cloud Infrastructure CLI commands have one of two common prefixes:

- akm: Use only for access key related commands.

  ```
  akm-create-access-key
  ```

- iaas: Use for all others cloud computing management commands.

  ```
  iaas-create-volume
  ```

Every Cloud Infrastructure CLI command have some common options. Depending on the command prefix options vary. The following table list the command options, description, and indicate if it is a required for the command.

Command Type	Option	Description	Required
akm and iaas	--base-url `<base_url>`	Base URL of the Enterprise Controller where the Web service is running. `https://<EnterpriseController host>`.	Yes
akm	--user	The username for authentication.	Yes
akm	--password-file|-p `<pw_file>`	Path to the password file for authentication. If this option is not supplied, the cloud user will instead be prompted for a password interactively	No
iaas	--access-key-file|-a `<access_key_file>`*	Path of the file storing the access key.	Yes
akm and iaas	--help|-h	Explains the command usage and its arguments.	No
akm and iaas	--header|-H	Adds a header row to the output.	No
akm and iaas	--sep `<separator>`	Specifies a column separator character. he default separator is TAB.	No

Command Type	Option	Description	Required
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 the common required options can also be specified via 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. There is also a short version for each command. Some command options also have a short version.

The following CLI commands are described in this section:

Functionality	Commands
Account Access	akm-create-access-key, akm-delete-access-key, akm-describe-access-keys, akm-describe-accounts
Server Template Management	iaas-create-server-template-from-assembly, iaas-create-server-template-from-url, iaas-create-server-template-from-vserver, iaas-delete-server-template, iaas-describe-server-templates
Virtual Network Management	iaas-create-vnet, iaas-delete-vnet, iaas-describe-vnets
vIP Address Management	iaas-allocate-ip-addresses, iaas-describe-ip-addresses, iaas-release-ip-addresses
Volume Management	iaas-attach-volumes-to-vserver, iaas-create-volume, iaas-delete-volume, iaas-describe-volumes, iaas-detach-volumes-from-vserver, iaas-import-volume
Snapshot Management	iaas-create-snapshot, iaas-delete-snapshot, iaas-describe-snapshots
Key Pair Management	iaas-create-key-pair, iaas-delete-key-pair, iaas-describe-key-pairs, iaas-import-key-pair
vServer Management	iaas-describe-vservers, iaas-describe-vserver-types, 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
Distribution Group Management	iaas-create-distribution-group, iaas-delete-distribution-group, iaas-describe-distribution-groups
Resource Attribute Management	iaas-describe-attributes, iaas-modify-attributes
Tag Management	iaas-create-tags, iaas-delete-tags, iaas-describe-tags
vDC Capabilities	iaas-describe-vdc-capabilities

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 <user name>] [--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, 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 verify the cloud user has the correct privileges for the specified account. If this option is not specified, 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 <pw_file>
Complete path and name of the file containing the password of the cloud user, for example /tmp/akpsswd.file. If a path is not specified for this file, 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, the cloud user is prompted for a password interactively.

--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>
Username of another cloud user to create an access key on behalf of that 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 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 the access key with the ID *AK_3*. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
user	<username>
password-file	~/pwd.file
account	ACC-e7ed508a-dad3-4866-a41a-bf8850163b2e
access-key-file	ak.file

```
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. This example uses these option values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/

Option	Environment Variable	Environment Variable Value
user	IAAS_USER	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file
account	N/A	ACC-g7fg508a-dad3-4866-a41a-bf8850163c3f
access-key-file	N/A	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*. This example uses these option values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
user	IAAS_USER	cloudadmin
for-user	N/A	<username>1
password-file	IAAS_PASSWORD_FILE	~/pwd.file
account	N/A	ACC-g7fg508a-dad3-4866-a41a-bf8850163c3f
access-key-file	N/A	ak.file

```
bash-3.2$ ./akm-create-access-key --account
ACC-g7fg508a-dad3-4866-a41a-bf8850163c3f --access-key-file ak.file --for-user
<username>1

AK_10
bash-3.2$
```

Exit Status

(Success) The command returns the access key ID of the newly created access key. When using the akm-describe-access-keys command, the access key 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

An access key is unregistered for 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 using 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, 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 verify the cloud user has the correct privileges for the specified account. If this option is not specified, 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 <pw_file>
Name of the file containing the password of the cloud user. If a path is not specified for this file, 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, the cloud user is prompted for a password interactively.

--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. Note. Access key IDs are displayed in the UI or by using akm-describe-access-keys command.

Examples

Example 1
This example unregisters an access key. This example uses these 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. This example uses these options and values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
user	IAAS_USER	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file
access-key-id	N/A	AK_4

```
bash-3.2$ ./akm-delete-access-key AK_4

bash-3.2$
```

Exit Status

(Success) The command does not return any value, just the command prompt is returned. When using the akm-describe-access-keys command, 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>]
                                  [--xml]
```

Description

This command list information of the access keys created for a cloud user. The command displays the access key ID, the account, and username 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, 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 verify the cloud user has the correct privileges for the specified account. If this option is not specified, 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 <pw_file>
Name of the file containing the password of the cloud user. If a path is not specified for this file, 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, the cloud user is prompted for a password interactively.

--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>
Username of a different cloud user to display the access keys created for that cloud user. This option validates that the cloud user using this option has 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. This example uses these 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. This example uses these options and values:

Option	Environment Variable	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 accounts 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 username 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, 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 verify the cloud user has the correct privileges for the specified account. If this option is not specified, 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 <pw_file>
Name of the file containing the password of the cloud user. If a path is not specified for this file, 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, the cloud user is prompted for a password interactively.

--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>

Username 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 information displayed to the specified account.Note. account IDs are displayed in the UI or by using akm-describe-user-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 displays the accounts information for a cloud user. This example uses these 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 accounts information for a different cloud user including the header row. This example uses these options and values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
user	IAAS_USER	<username>
password-file	IAAS_PASSWORD_FILE	~/pwd.file
for-user	N/A	<username>0

```
bash-3.2$ ./akm-describe-accounts --for-user <username>0 -H

account name    description     user
```

```
ACC-f7gd508a-dad3-4866-a41a-bf8850163c3g        account0        account0 for user
<username>0        <username>
ACC-95gdb073-d569-4a45-ac24-f14eba689a8g        account1        account1 for user
<username>0        <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 could 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, 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, 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 values 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. This example uses these option 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. This example uses these option 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, these 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

Attach one or more volumes to a stopped 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 stopped 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, 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, 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 attachs a volume to a vServer. This example uses these option 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-0fb57293-347c-4717-96ef-6dd23154596f  --volume-ids
VOL-052cb4b4-5e56-4303-8b3a-82d6ba743a15

bash-3.2$
```

Exit Status

(Success) The command attachs the specified volume to the vServer. This command doesn't return any other response.

(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

Create 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>]
                                [--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, 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, 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`.

--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. This example uses these option 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$
```

Exit Status

(Success) The command returns the distribution group ID. When using the
iaas-describe-distribution-groups command, the distribution group is listed.

and name? per demo

(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, 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, 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, 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. This example uses these option 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 key pair is listed.

and name? per demo

(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

Create 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, 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, 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, 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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myAssembly
url	http://myServer.com/vm-templates/OVM.tar.gz

```
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 using the iaas-describe-server-templates command, the 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 in 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, 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, 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.

Note: in demo, default description comes from OVM/third party?

--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. This example uses these option 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

Create 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, 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, 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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
name	myST
vServer ID	http://myServer.com/vm-templates/OVM.tar.gz

```
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

Create 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, 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, 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 of the specified volume ID. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
volumeID	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 of the specified volume ID. This example uses these option values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	ak.file
volumeID	N/A	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, 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, 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 resource. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
resource 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 resource. This example uses these option 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, just the command prompt. When using the iaas-describe-tags command, the tag is listed.

(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, 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, 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. The default value is.

--desc|-d <descr>
Description of t he 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. This example uses these option 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. This uses these option values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	ak.file
name	N/A	myVNET2
size	N/A	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 a the vNet ID of the vNet created. When using the iaas-describe-vnets command, the vNet is listed.

(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 have also the 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, 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, 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 non-shared volume.

--size|-s <size>
Size of the volume in 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 a empty volume of 2 GB. This example uses these option 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 base on the specified snapshot. This uses these option values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	<username>
snapshot ID	N/A	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 a the volume ID of the volume created. When using the iaas-describe-volumes command, the volume is listed

(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

Delete 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, 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, 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. This example uses these option 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, just 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-vserver
- 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, 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, 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 in an account. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key pair 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, just 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 a server template based on 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, 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, 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. This example uses these option 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, just 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, 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, 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. This example uses these option 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, just 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

Delete 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 using the tag name.

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, 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, 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>
Specify the column separator character(s), 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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
resource ID	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t
tag name	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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
resource ID	VNET-5d74972a-bcdd-4714-8c7f-b67d8010f25t
tag1 name	myTag
tag2 name	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, just the command prompt.

(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 an 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, 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, 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. This example uses these option 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-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 message, the response is simply empty.

(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

Delete 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, 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, 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. This example uses these option 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 message, the response is simply empty.

(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, 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, 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 output of a 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. This example uses
these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
IvNet 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 attribute of the specified server template.
This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
server template ID	TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01

```
bash-3.2$ ./iaas-describe-attributes --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --id
TMPL-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01 --attr-names name,status

name    default-server-template1
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, 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, 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>
Specify the column separator character(s), 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 distribution groups in an account. This example uses these option 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
DG-8e7b0d64-57ca-492d-806d-a2db6b9a00df distribGP          OK
DG-068ae84c-d0fc-406d-aa37-0be4f88d411c myDistributionGroup          OK
bash-3.2$
```

Exit Status

(Success) The command returns the list of distribution groups and their attributes. If no distribution groups are found, the response is simply 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 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 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 in case 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, 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, 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>
Specify the column separator character(s), 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. This example uses these option 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-d917712abaaa192.168.2.1
VNET-vfr375904-9d34-4ec8-86b9-d917712abaaa
bash-3.2$
```

Example 2

This example lists the allocated IP addresses from the specified virtual network. This example uses these option values:

Option	Option value
base-url	https://\<EnterpriseControllerHostname\>/
access-key-file	ak.file
vNet ID	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,vnet=VNET-cde375904-9d34-4ec8-86b9-
d917712abaaa

192.168.2.5    VNET-cde375904-9d34-4ec8-86b9-d917712abaaa
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, the response is simply 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 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, 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, 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. This example uses these option 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. This example uses these option 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, the response is simply 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-vserver
- iaas-run-vservers

iaas-describe-server-templates

Lists server templates for an account and their attributes.

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 loaded 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, 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, 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 adding the header row. This example uses these option 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. This uses these option values:

Option	Environment Variable	Environment Variable Value
base-url	IAAS_BASE_URL	https://<EnterpriseControllerHostname>/
access-key-file	IAAS_ACCESS_KEY_FILE	<username>
filters	N/A	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 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. This 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, 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, 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. This example uses these option 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 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. This 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, 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, 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, entity type (instance, template, key pair, vnet, volume, snapshot or instance snapshot) and entity identifier. The result contains only tags that matches against all elements of the filter list.

--ids|-i <res-id>[,<res-id>]*
IDs of resources in the account to display only the tags associated to these resources.

--sep <separator>
Specify the column separator character(s), 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 to them in an account along
with the tag name and value. This example uses these option 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. This
example uses these option 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-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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
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 vDC capabilities for an account. This command 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, 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, 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. This example uses these option 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
DynamicVNetCreation     disabled
DistributionGroupSupport        disabled
bash-3.2$
```

Example 2

This example lists only the virtualization type of the vDC for an account. This example uses these option 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-describe-vnets

Lists virtual networks 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 list the available virtual networks in an account. This command 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, 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, 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 matches against 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. This example uses these option 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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
vNet ID	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 simply 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 volumes 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 available volumes for an account. This command also allows filtering of 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, 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, 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, and shared.

--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. This example uses these option 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

VOL-7f7b1963-283c-4f2d-9748-30cf87119cef myVol1 DescmyVol1 OK 0
VOL-82b4661e-de12-4931-b9a4-cce8edd23532 myVol2 DescmyVol1 OK 8
bash-3.2$
```

Example 2
This example lists the attributes of the specified volume IDs. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
volume 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 0
VOL-82b4661e-de12-4931-b9a4-cce8edd23532 myVol2 DescmyVol1 OK 8
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 simply 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-vservers

Displays vServers 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 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, 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, 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 <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. This example uses these option 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
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee05       name modified  Default Virtual
Server  OK
VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff01,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff0
2,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff03,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff
f04,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff05          small
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee03       default-vserver3        Default
Virtual Server  OK
VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff01,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff0
2,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff03,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff
f04,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff05   mockupKey       small
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee02       default-vserver2        Default
Virtual Server  OK
VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff01,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff0
2,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff03,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff
f04,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff05   mockupKey       small
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee04       default-vserver4        Default
Virtual Server  OK
VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff01,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff0
2,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff03,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff
f04,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff05   mockupKey       small
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee01       default-vserver1        Default
Virtual Server  OK
VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff01,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff0
2,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff03,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff
f04,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff05   mockupKey       small
bash-3.2$
```

Example 2

This example lists the attributes of the specified vServer IDs. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file

Option	Option value
vServer IDs	VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee02
	VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee03

```
bash-3.2$./iaas-describe-vservers --base-url
https://<EnterpriseControllerHostname>/ -a ak.file --ids
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee02,VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee0
3 -H

id      name     description    status vnets    ipAddresses      serverTemplateId
keyName vserverType
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee03       default-vserver3       Default
Virtual Server  OK
VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff01,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff0
2,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff03,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff
f04,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff05      mockupKey       small
VSRV-aaaaaaa8-bbb4-ccc4-ddd4-eeeeeeeeee02       default-vserver2       Default
Virtual Server  OK
VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff01,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff0
2,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff03,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff
f04,VNET-aaaaaaa8-bbb4-ccc4-ddd4-ffffffffff05      mockupKey       small
bash-3.2$
```

Exit Status

(Success) The command returns the list of existing vServers and their attributes. If no volumes are found, the response is simply empty.

(Failure) The command displays a descriptive error message and error code.

See Also

- 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 available vServers 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 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, 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, 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. This example uses these option 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 vSever types and their attributes. If no vServer types are found, the response is simply empty.

(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

Detach one or more volumes from a stopped 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 stopped 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, 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, 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.

--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 a volume from a vServer. This example uses these option 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-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 volume from the vServer. This command doesn't return any other response.

(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, 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, 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. This example uses
these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
file containing a public key	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-vserver
- 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, 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, 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. This
example uses these option 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 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 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, 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, 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 using the corresponding iaas-describe* command, for example iaas-describe-server-template.

--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 a 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 modifies the name attribute of the specified vNet. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
IvNet 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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
Iserver template 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, just the command prompt.

(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, 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, 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. This example uses these option 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-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 doesn't return any other response.

(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 one 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 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, 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, 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 sends a message to a vServer. This example uses these option 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-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-release-ip-addresses

Release 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 release 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, 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, 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. This example uses these option 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-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 3 allocated IP addresses of the specified virtual network. This example uses these option 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-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 returns 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>
                          [--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, 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, 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|-k <key-pair-name>
Name of the 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 volumeIDs.

--vserver-type <type>]
vServer type.

--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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key pair 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

Create 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>
                            [--sep <separator>]
                            [--trust-store <truststore_file>]
                            [--verbose] [--xml]
```

Description

This command create and starts one or more vServers. A suffix is appended a numerical sequence to the name to distinguish among all vServers created.

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, 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, 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, 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.

--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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
key pair	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 OS. The message is sent to a common buffer in the virtualization layer that have 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.

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, 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, 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. This example uses these option 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 doesn't return any other response.

(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, 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, 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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
IvServer 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, just the command prompt.

(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, 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, 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. This example uses these option values:

Option	Option value
base-url	https://<EnterpriseControllerHostname>/
access-key-file	ak.file
IvServer 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 value, just the command prompt.

(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, 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, the command looks for the file in the current directory.

--help|-h
Displays the usage information for this command.

--force|f
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. This example uses these option 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 doesn't return any other response.

(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

A

A Web Service Client Example

This appendix provides an example for generating requests to the Web Service using the WebUtil application. The appendix includes:

- **Description of Web Service API Requests**
- **Sending a Web Service Request**
- **Creating Public and Private Keys**
- **WebUtil Application Overview**
- **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 AKM request to the Web Service API:

https://USER:PASSWORD@HOST/akm/?REQUEST_DATA

Name	Description
USER	Cloud user name
PASSWORD	Cloud user password
HOST	IP address or fully qualified hostname
REQUEST_DATA	The request data is made of a series of name=value parameters separated by "&"

REQUEST_DATA

The parameters required as part of the REQUEST_DATA depend on the value used for the AKM action. However, there are a set of common parameters required for every AKM request.

The following table shows the common required parameters for the REQUEST_DATA in every AKM request:

Name	Description
Action	One of the akm action. See, for a complete list of akm actions.
Version	1
Timestamp	Timestamp in milliseconds since 1st of January 1970.
Expires	Expiry date in milliseconds since 1st of January 1970, must be greater than the timestamp field above. A 300000ms difference between the 2 values should be more than enough.

See RegisterAccessKeyRequest, DescribeAccessKeysRequest, DeleteAccessKeyRequest, and DescribeAccountsRequest for a complete description and extra parameters that can be used for the AKM actions.

Example of a complete 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 IAAS request using the Web Service API:

```
https://HOST/iaas/?REQUEST_IAAS_DATA&SIGNATURE_BLOCK
```

Name	Description
HOST	IP address or fully qualified hostname of Ops Center
REQUEST_IAAS_DATA	The request data is made of a series of name=value parameters separated by "&"
SIGNATURE_BLOCK	The signature block is a series of name=value parameters separated by "&"

REQUEST_IAAS_DATA

The parameters required as part of the REQUEST_IAAS_DATA depend on the value used for the IAAS action. However, there are a set of common parameters required for every IAAS request.

The following table shows the common required parameters for the REQUEST_IAAS_DATA in every IAAS request:

Name	Description
Action	One of the IAAS actions. See, for a complete list of the IAAS actions.
Version	1
Timestamp	Timestamp in milliseconds since 1st of January 1970

Name	Description
Expires	Expiry date in milliseconds since 1st of January 1970, must be greater than the timestamp field above. A 300000ms difference between the 2 values should be more than enough.
accessKeyId	The value returned by the akm RegisterAccessKey action.

See List of Actions, for a complete list of IAAS actions and extra parameters that can be used.

Example A–1   An Example of REQUEST_IAAS_DATA

```
Action=
DescribeVnets&Version=1&Timestamp=1330954619299&Expires=1330954919299&accessKeyId=
AK_1
```

SIGNATURE_BLOCK

The signature block consists of signing some IAAS data with a private key so the web service can authenticate the request. The web server should be able to verify the signature with the public key which has been registered with the RegisterAccessKey action.

The following table shows the series of name=value parameters that conform the SIGNATURE_BLOCK. These parameter are separated by "&":

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–2   An Example of SIGNATURE_BLOCK

```
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=bj8GfJCqvPZZPU2JoWAGzZd
CF+N767rQejILMQwNdgKLfoGGqAwDPRYMr/ghUoBc6RB3nKYgAyPdmtCfhzRGTqECgUWy0jCrE99+utGee
J0/XRQ9LxyYeBgzjO3lHP+hFhUo+gUtQaSYPhUHH7eTkxg/CrolMxibglypJM/rIf90yEqSeqhphQt7hWx
lT0DNAy6/cZt8isT/Tu8V7ZFjBFkEpLfN97bIOJ2vIIpOeetmftuw4ObtqjbUp6+7dpVkhhCQnX0MAIDj+
mjorEOzcwK+F1pYuzES0fjaW0MowG+cA/9gttDjg7r5H29i3qbbjIlvAt6fk1HPpSxQTSTOTg==
```

IAAS Data Used for Signature

To generate the encoded hash value of the SIGNATURE_BLOCK for a IAAS request, the following data is used, this data needs to be signed and base64 encoded:

Name	Description
Http Request type	POST.
HostName	IP address or fully qualified hostname of Ops Center)
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 "&". See also REQUEST_IAAS_DATA.

Example A–3   Example of the IAAS data to sign

Note that there is a `"\n"` between every field.

```
POST
<EnterpriseControllerHostname>
/iaas/
Action= DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1330954919299
```

Example A–4   Java representation of the data to sign

Note that there is a `"\n"` between every field.

```
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 A Web Service Client Example, for an example of generating IAAS request with signed data.

Sending a Web Service Request

Request to the Web Service API are simple web POST or GET operations. Once 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 WGET.

Using the Web Browser

To use the web browser to send a request to the Web service API, 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

To use WGET, 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

A public and private RSA keys are required for signing the required IAAS data so that the web service can authenticate the request.

Private Key

Before creating a public key a private key must be created. A private key can be created executing the following command:

```
openssl genrsa -out privatekey.pem 2048
```

The private key is stored in the *privatekey.pem* file and this file is used to create the public key. Once the public key is registered with the RegisterAccessKey action, this private key can be used to sign IAAS data.

To use WebTuil to sign IAAS data, the private key must be stored in DER format, this can be achieved by using the following command:

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in privatekey.pem -nocrypt >
privatekey.DER
```

The private key is stored in *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 and can be used with the RegisterAccessKey action so that the web server can verify the signed data with the public key.

WebUtil Application Overview

WebUtil is a simple java application that generates the signature for an IAAS request to the Web Service API. WebUtil uses the SHA512withRSA method and base64 encode to signed the required data.

The following is the code of the WebUtil application. When building the application with the `javac` compiler it is possible that 2 warnings are being displayed.

```
/*
 * 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

Before using the WebUtil application an access key for an account needs to be created. See Creating an Access Key for more information.

Once the access key is created, the IAAS data must be defined, see also IAAS Data Used for Signature.

The following is an example of this data for viewing vNets information.:

```
POST
<EnterpriseControllerHostname>
Action=DescribeVnets&Version=1&accessKeyId=AK_
1&Timestamp=12333333&Expires=13333444
```

To generate the signature and base64 encoded 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 *signedData*:

*bj8GfJCqvPZZPU2JoWAGzZdCF+N767rQejILMQwNdgKLfoGGqAwDPRYMr/ghUoBc6RB3nKYgAy
PdmtCfhzRGTqECgUWy0jCrE99+utGeeJ0/XRQ9LxyYeBgzjO3lHP+hFhUo+gUtQaSYPhUHH7eT
kxg/CrolMxibglypJM/rIf90yEqSeqhphQt7hWxlT0DNAy6/cZt8isT/Tu8V7ZFjBFkEpLfN97
bIOJ2vIIpOeetmftuw4ObtqjbUp6+7dpVkhhCQnX0MAIDj+mjorEOzcwK+F1pYuzES0fjaW0Mo
wG+cA/9gttDjg7r5H29i3qbbjIlvAt6fk1HPpSxQTSTOTg==*

WebUtil also generates the *iaasPartReques*t 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:

1. Creating an Access Key

2. 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 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/mE7b3gdA6d9VuwIPnZJ5ZFQUZCL
   yhaAotLCdACrzbgzYXdqt+rstutT1AVkE2UAMcm5O3KnIoObZKb8JtepSt74A9Rg
   VBkcCBjmKGfLNOL1KlZconkITm85TWKRaGRFuASxdl2ZrD723ZNb66X/a9ebxTMr
   6vVeskcaZpPlHzvgMOpiyDGwRvxn9yM5WB83zFDGT26Lihn/bKzLJXa+F2YNkLrT
   JQIDAQAB
   -----END PUBLIC KEY-----
   ```

3. Describe accounts for a cloud user.

   a. Create and send an AKM request using the DescribeAccounts action. The appropriate hostname, cloud user, timestamp and expires information must be filled properly:

      ```
      https://<username>:<password>@<EnterpriseControllerHostname>/akm/?
      Action=DescribeAccounts&Version=1&Timestamp=1330954619299&Expires=1
      330954919299
      ```

b. Choose the account ID from the result returned from the previous AKM request, for example: 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 previously created 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 previously:

https://<username>:<password>@<EnterpriseControllerHostname>/akm/?Action=RegisterAccessKey&Version=1&Timestamp=1330975344&Expires=1333975344&account=ACC-0162da5a-5d25-4096-af59-3dd1de27cfad&publicKey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAufVdjdp0MmOLbNypLVMWXfmhusawid4Wg4n4FZewSmoBEYA8f8wIA0SI87Shi7RtMcWsEoXvNNHA0wcJoA1RjyVLsI3rtrq0c0k7AxQSwb4UK/rSXW1NXxMh/mE7b3gdA6d9VuwIPnZJ5ZFQUZCLyhaAotLCdACrzbgzYXdqt+rstutT1AVkE2UAMcm5O3KnIoObZKb8JtepSt74A9RgVBkcCBjmKGfLNOL1KlZconkITm85TWKRaGRFuASxdl2ZrD723ZNb66X/a9ebxTMr6vVeskcaZpPlHzvgMOpiyDGwRvxn9yM5WB83zFDGT26Lihn/bKzLJXa+F2YNkLrTJQIDAQAB

b. Get the access key ID generated. The access key ID is part of the results returned from the RegisterAccessKey action. This access key ID is used for all next IAAS request. For this example: 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:

■ Describe vNets

■ Describe vNets filtering by vNet ids

■ Create a private vNet

■ Describe a Vnet using filters

To execute these actions using the Web Service, the access key ID and private key file obtained from previous steps is used.

Describe vNets

1. Define the data to be signed using DescribeVnets action:

Action=DescribeVnets&Version=1&AccessKeyId=AK_1

2. Sign the data using the WebUtil application.

```
java WebUtil signature privatekey.DER "POST" "<EnterpriseControllerHostname>"
"Action=DescribeVnets&Version=1&accessKeyId=AK_1" signedData
```

3. Get the signed data stored in *iaasPartRequest* and *signedData* files.

Content of `iaasPartRequest`:
```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Versi
on=1&AccessKeyId=AK_
1&Timestamp=1331058169938&Expires=1331058469938&SignatureMethod=SHA512w
ithRSA&SignatureVersion=1&Signature=
```

Content of *signedData*:
bj8GfJCqvPZZPU2JoWAGzZdCF+N767rQejILMQwNdgKLfoGGqAwDPRYMr/ghUoBc6RB3nKY
gAyPdmtCfhzRGTqECgUWy0jCrE99+utGeeJ0/XRQ9LxyYeBgzjO3lHP+hFhUo+gUtQaSYPh
UHH7eTkxg/CrolMxibglypJM/rIf90yEqSeqhphQt7hWxlT0DNAy6/cZt8isT/Tu8V7ZFjB
FkEpLfN97bIOJ2vIIpOeetmftuw4ObtqjbUp6+7dpVkhhCQnX0MAIDj+mjorEOzcwK+F1pY
uzES0fjaW0MowG+cA/9gttDjg7r5H29i3qbbjIlvAt6fk1HPpSxQTSTOTg==

4. Append the signature to get the complete IAAS request, for example:

```
https://<EnterpriseControllerHostname>/iaas/?Action=DescribeVnets&Versi
on=1&AccessKeyId=AK_
1&Timestamp=1331058169938&Expires=1331058469938&SignatureMethod=SHA512w
ithRSA&SignatureVersion=1&Signature=bj8GfJCqvPZZPU2JoWAGzZdCF+N767rQejI
LMQwNdgKLfoGGqAwDPRYMr/ghUoBc6RB3nKYgAyPdmtCfhzRGTqECgUWy0jCrE99+utGeeJ
0/XRQ9LxyYeBgzjO3lHP+hFhUo+gUtQaSYPhUHH7eTkxg/CrolMxibglypJM/rIf90yEqSe
qhphQt7hWxlT0DNAy6/cZt8isT/Tu8V7ZFjBFkEpLfN97bIOJ2vIIpOeetmftuw4ObtqjbU
p6+7dpVkhhCQnX0MAIDj+mjorEOzcwK+F1pYuzES0fjaW0MowG+cA/9gttDjg7r5H29i3qb
bjIlvAt6fk1HPpSxQTSTOTg==
```

> **Note:** The complete request to the Web service must be created as
> single line request.

5. Send the complete IAAS request using the browser or wget. The result returned is
   similar to:

```xml
<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 OVM Discovered Network </description>
<status>OK</status>
<ipAddress>10.169.69.0/24</ipAddress>
</items>
</result>
```

Describe vNets filtering by vNet ids

1. Define the data to be signed using DescribeVnets action:

```
Action=DescribeVnets&Version=1&AccessKeyId=AK_
1&ids.1=VNET-6c774d08-d0a7-493b-9fa1-c93a80702f8d
```

2. Sign the data 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 stored in *iaasPartRequest* and *signedData* files.

Content of `iaasPartRequest`:
```
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 *signedData*:
*B15A5WdQkIwTrlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw23sHBdRvhzLuh
yW2LWm90EIhZHd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxfray6eSGCcpHQW
XaA+TjpCitzguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmhH7iI7pBC8gcpomnDZhx
ZkAFaVT0TFf88epbJal5Yl6aTpptjlUNEz6jDhuV2Qaa3bI5OqnJc0PJgCkhuKU7un8us1J
VEk/nf/or5RCBMmPGAluuILIfYBEe/euVUy7wQ00eyKPRuplZkORmVbFLg==*

4. Append the signature to get the complete IAAS request, for 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=B15A5WdQkIwTrlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw
23sHBdRvhzLuhyW2LWm90EIhZHd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxf
ray6eSGCcpHQWXaA+TjpCitzguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmhH7iI7p
BC8gcpomnDZhxZkAFaVT0TFf88epbJal5Yl6aTpptjlUNEz6jDhuV2Qaa3bI5OqnJc0PJgC
khuKU7un8us1JVEk/nf/or5RCBMmPGAluuILIfYBEe/euVUy7wQ00eyKPRuplZkORmVbFLg
==*

> **Note:** The complete request to the Web service must be created as single line request.

5. Send the complete IAAS request using the browser or wget. The result returned is similar to:

```
<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 OVM Discovered Network </description>
<status>OK</status>
<ipAddress>10.169.69.0/24</ipAddress>
</items>
</result>
```

Create a private vNet

1. Define the data to be signed using DescribeVnets action:

Action=CreateVnet&Version=1&AccessKeyId=AK_1&name=privatevnetWebApi

2. Sign the data 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 stored in *iaasPartRequest* and *signedData* files.

Content of `iaasPartRequest`:
*https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=
1&AccessKeyId=AK_
1&name=privatevnetWebApi&Timestamp=1331058639019&Expires=1331058939019&
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=*

Content of *signedData*:

B15A5WdQkIwTrlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw23sHBdRvhzLuh
yW2LWm90EIhZHd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxfray6eSGCcpHQW
XaA+TjpCitzguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmhH7iI7pBC8gcpomnDZhx
ZkAFaVT0TFf88epbJal5Yl6aTpptjlUNEz6jDhuV2Qaa3bI5OqnJc0PJgCkhuKU7un8us1J
VEk/nf/or5RCBMmPGAluuILIfYBEe/euVUy7wQ00eyKPRuplZkORmVbFLg==

4. Append the signature to get the complete IAAS request, for example:

*https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=
1&AccessKeyId=AK_
1&name=privatevnetWebApi&Timestamp=1331058639019&Expires=1331058939019&
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=B15A5WdQkIwT
rlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw23sHBdRvhzLuhyW2LWm90EIhZ
Hd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxfray6eSGCcpHQWXaA+TjpCitzg
uoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmhH7iI7pBC8gcpomnDZhxZkAFaVT0TFf8
8epbJal5Yl6aTpptjlUNEz6jDhuV2Qaa3bI5OqnJc0PJgCkhuKU7un8us1JVEk/nf/or5RC
BMmPGAluuILIfYBEe/euVUy7wQ00eyKPRuplZkORmVbFLg==*

> **Note:** The complete request to the Web service must be created as single line request.

5. Send the complete IAAS request using the browser or wget. The result returned is similar to:

```
<result xsi:type="CreateVnetResult" requestId="109">
<vnetId>VNET-7e33ee2e-d0f9-4b39-b8c9-32b8c8bfed87</vnetId>
</result>
```

Describe a Vnet using filters

1. Define the data to be signed using DescribeVnets action:

*Action=DescribeVnets&Version=1&AccessKeyId=AK_
1&filters.1.filterValue=privatevnet1&filters.1.filterName=name*

2. Sign the data 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 stored in *iaasPartRequest* and *signedData* files.

Content of iaasPartRequest:
*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 *signedData*:
*B15A5WdQkIwTrlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw23sHBdRvhzLuh
yW2LWm90EIhZHd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxfray6eSGCcpHQW
XaA+TjpCitzguoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmhH7iI7pBC8gcpomnDZhx
ZkAFaVT0TFf88epbJal5Yl6aTpptjlUNEz6jDhuV2Qaa3bI5OqnJc0PJgCkhuKU7un8us1J
VEk/nf/or5RCBMmPGAluuILIfYBEe/euVUy7wQ00eyKPRuplZkORmVbFLg==*

4. Append the signature to get the complete IAAS request, for example:

```
https://<EnterpriseControllerHostname>/iaas/?Action=CreateVnet&Version=
1&AccessKeyId=AK_
1&name=privatevnetWebApi&Timestamp=1331058639019&Expires=1331058939019&
SignatureMethod=SHA512withRSA&SignatureVersion=1&Signature=B15A5WdQkIwT
rlR/NDXbn1RFwWyBFAQFtZTfZ1jH8ftkRgvfwU93nxukpw23sHBdRvhzLuhyW2LWm90EIhZ
Hd3H2x9s8D1wCAUFTsSAZoPuIlW40qn0sf9VTEXOlqwnxfray6eSGCcpHQWXaA+TjpCitzg
uoPmHa3LjGKECnbNuqJS15L8VuVfE6otaxyTBzmhH7iI7pBC8gcpomnDZhxZkAFaVT0TFf8
8epbJal5Yl6aTpptjlUNEz6jDhuV2Qaa3bI5OqnJc0PJgCkhuKU7un8us1JVEk/nf/or5RC
BMmPGAluuILIfYBEe/euVUy7wQ00eyKPRuplZkORmVbFLg==
```

> **Note:** The complete request to the Web service must be created as single line request.

5. Send the complete IAAS request using the browser or wget. The result returned is similar to:

```
<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>
```