



Document Number: TBD

Date: 2010-06-28

Version:1.0

1
2
3
4

5 Oracle Cloud Resource Model API

6 **Document Type: Specification**
7 **Document Status: Open for Comments**
8 **Document Language: en-US**
9 **Document Version: 0.34**

10 Copyright Notice
11 Copyright © 2010 Oracle. All rights reserved.

12

ORACLE TECHNOLOGY NETWORK

13

DEVELOPMENT AND DISTRIBUTION LICENSE AGREEMENT

14

“We,” “us,” and “our” refers to Oracle USA, Inc., for and on behalf of itself and its subsidiaries and affiliates under common control. “You” and “your” refers to the individual or entity that wishes to use the specification from Oracle. “Specification” refers to the specification you wish to download and use.

16

“License” refers to your right to use the specification under the terms of this agreement. This agreement is governed by the substantive and procedural laws of California. You and Oracle agree to submit to the exclusive jurisdiction of, and venue in, the courts of San Francisco, San Mateo, or Santa Clara counties in California in any dispute arising out of or relating to this agreement.

17

18

19

20

21

22

We are willing to license the specification to you only upon the condition that you accept all of the terms contained in this agreement. Read the terms carefully and select the “Accept” button at the bottom of the page to confirm your acceptance. If you are not willing to be bound by these terms, select the “Do Not Accept” button and the registration process will not continue.

23

24

25

26

27

License Rights

28

We grant you a nonexclusive, nontransferable limited license to use the specification for purposes of developing your applications. You may also distribute the specification with your applications to your customers. If you want to use the specification for any purpose other than as expressly permitted under this agreement you must contact us to obtain the appropriate license. We may audit your use of the specification.

29

30

31

32

33

34

Ownership and Restrictions

35

We retain all ownership and intellectual property rights in the specification. You may make a sufficient number of copies of the specification for the licensed use.

36

37

38

You may not:

39

- use the specification for any purpose other than as provided above;
- distribute the specification unless accompanied with your applications;
- charge your end users for use of the specification;
- remove or modify any program markings or any notice of our proprietary rights;
- use the specification to provide third party training on the content and/or functionality of the specification, except for training your licensed users;
- assign this agreement or give the specification, program access or an interest in the specification to any individual or entity except as provided under this agreement;
- disclose results of any program benchmark tests without our prior consent; or,
- use any Oracle name, trademark or logo.

40

41

42

43

44

45

46

47

48

49

50

Specification Distribution

51 We grant you a nonexclusive, nontransferable right to copy and distribute the specification to your end
52 users provided that you do not charge your end users for use of the specification and provided your end
53 users may only use the specification to run your applications for their business operations. Prior to
54 distributing the specification you shall require your end users to execute an agreement binding them to
55 terms consistent with those contained in this section and the sections of this agreement entitled "License
56 Rights," "Ownership and Restrictions," "Export," "Disclaimer of Warranties and Exclusive Remedies," "No
57 Technical Support," "End of Agreement," "Relationship Between the Parties," and "Open Source." You
58 must also include a provision stating that your end users shall have no right to distribute the specification,
59 and a provision specifying us as a third party beneficiary of the agreement. You are responsible for
60 obtaining these agreements with your end users.

61
62 You agree to: (a) defend and indemnify us against all claims and damages caused by your distribution of
63 the specification in breach of this agreements and/or failure to include the required contractual provisions
64 in your end user agreement as stated above; (b) keep executed end user agreements and records of end
65 user information including name, address, date of distribution and identity of specification distributed; (c)
66 allow us to inspect your end user agreements and records upon request; and, (d) enforce the terms of
67 your end user agreements so as to effect a timely cure of any end user breach, and to notify us of any
68 breach of the terms.

69

70 **Feedback**

71 You may provide us with comprehensive information and feedback regarding the specification, including,
72 but not limited to, information regarding any problems that you experience with the specification and/or
73 any improvements or suggested changes to the specification, and any such feedback or other information
74 shall be received and treated by us on a non-confidential and unrestricted basis ("Feedback"). Such
75 Feedback will be addressed at our discretion and according to our standard development practices. We
76 shall have a worldwide, royalty-free, non-exclusive, perpetual, and irrevocable right to use Feedback for
77 any purpose, including but not limited to, incorporation of such Feedback into the specification or other
78 software products.

79

80 **Export**

81 You agree that U.S. export control laws and other applicable export and import laws govern your use of
82 the specification, including technical data; additional information can be found on Oracle's Global Trade
83 Compliance web site located at <http://www.oracle.com/products/export/index.html?content.html>. You
84 agree that neither the specification nor any direct product thereof will be exported, directly, or indirectly, in
85 violation of these laws, or will be used for any purpose prohibited by these laws including, without
86 limitation, nuclear, chemical, or biological weapons proliferation.

87

88 **Disclaimer of Warranty and Exclusive Remedies**

89

90 **THE SPECIFICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. WE FURTHER**
91 **DISCLAIM ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION, ANY**
92 **IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR**
93 **NONINFRINGEMENT.**

94

95 **IN NO EVENT SHALL WE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE OR**
96 **CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR DATA**
97 **USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR**
98 **TORT, EVEN IF WE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. OUR**
99 **ENTIRE LIABILITY FOR DAMAGES HEREUNDER SHALL IN NO EVENT EXCEED ONE THOUSAND**
100 **DOLLARS (U.S. \$1,000).**

101

102 **No Technical Support**

103 Our technical support organization will not provide technical support, phone support, or updates to you for
104 the specification licensed under this agreement.

105

106 **Restricted Rights**

107 If you distribute a license to the United States government, the specification, including documentation,
108 shall be considered commercial computer software and you will place a legend, in addition to applicable
109 copyright notices, on the documentation, and on the media label, substantially similar to the following:

110

NOTICE OF RESTRICTED RIGHTS

111 “Programs delivered subject to the DOD FAR Supplement are ‘commercial computer software’ and use,
112 duplication, and disclosure of the specification, including documentation, shall be subject to the licensing
113 restrictions set forth in the applicable Oracle license agreement. Otherwise, specification delivered
114 subject to the Federal Acquisition Regulations are ‘restricted computer software’ and use, duplication, and
115 disclosure of the specification, including documentation, shall be subject to the restrictions in FAR 52.227-
116 19, Commercial Computer Software-Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle
117 Parkway, Redwood City, CA 94065.”

118

119 **End of Agreement**

120 You may terminate this agreement by destroying all copies of the specification. We have the right to
121 terminate your right to use the specification if you fail to comply with any of the terms of this agreement, in
122 which case you shall destroy all copies of the specification.

123

124 **Relationship Between the Parties**

125 The relationship between you and us is that of licensee/licensor. Neither party will represent that it has
126 any authority to assume or create any obligation, express or implied, on behalf of the other party, nor to
127 represent the other party as agent, employee, franchisee, or in any other capacity. Nothing in this
128 agreement shall be construed to limit either party’s right to independently develop or distribute software
129 that is functionally similar to the other party’s products, so long as proprietary information of the other
130 party is not included in such software.

131

132 **Open Source**

133 “Open Source” software – software available without charge for use, modification and distribution – is
134 often licensed under terms that require the user to make the user’s modifications to the Open Source
135 software or any software that the user ‘combines’ with the Open Source software freely available in
136 source code form. If you use Open Source software in conjunction with the specification, you must
137 ensure that your use does not: (i) create, or purport to create, obligations of us with respect to the Oracle
138 specification; or (ii) grant, or purport to grant, to any third party any rights to or immunities under our
139 intellectual property or proprietary rights in the Oracle specification. For example, you may not develop a
140 software program using an Oracle program and an Open Source program where such use results in a
141 program file(s) that contains code from both the Oracle program and the Open Source program (including
142 without limitation libraries) if the Open Source program is licensed under a license that requires any
143 “modifications” be made freely available. You also may not combine the Oracle program with
144 specification licensed under the GNU General Public License (“GPL”) in any manner that could cause, or
145 could be interpreted or asserted to cause, the Oracle program or any modifications thereto to become
146 subject to the terms of the GPL.

147

148 **Entire Agreement**

149 You agree that this agreement is the complete agreement for the specification and licenses, and this
150 agreement supersedes all prior or contemporaneous agreements or representations. If any term of this
151 agreement is found to be invalid or unenforceable, the remaining provisions will remain effective.

152

Contents

154	Foreword	9
155	1 Scope	11
156	1.1 In-Scope	11
157	1.2 Out of Scope	11
158	1.3 Conformance	11
159	2 Normative References	11
160	3 Terms and Definitions	11
161	4 Symbols and Abbreviated Terms	13
162	5 Executive Summary	13
163	6 Introduction	13
164	7 Specification	14
165	8 Common Behaviors	15
166	8.1 Transport Protocol	15
167	8.2 URI Space	15
168	8.3 Media Types	15
169	8.4 Request Headers	16
170	8.5 Request Parameters	16
171	8.6 Response Headers	17
172	8.7 HTTP Status Codes	18
173	8.8 Common Resource Attributes	20
174	8.9 Error Response Message [application/vnd.com.oracle.cloud.common.Messages+json]	22
175	8.10 Extensibility to the resource model	24
176	9 Cloud Resource Models	24
177	9.1 Cloud [application/vnd.com.oracle.cloud.Cloud+json]	25
178	9.2 ServiceTemplate [application/vnd.com.oracle.cloud.ServiceTemplate+json]	27
179	9.3 VMTemplate [application/vnd.com.oracle.cloud.VMTemplate+json]	29
180	9.4 AssemblyTemplate [application/vnd.com.oracle.cloud.AssemblyTemplate+json]	32
181	9.5 Zone [application/vnd.com.oracle.cloud.Zone+json]	33
182	9.6 Location [application/vnd.com.oracle.cloud.Location+json]	34
183	9.7 VDC [application/vnd.com.oracle.cloud.VDC+json]	35
184	9.8 AssemblyInstance [application/vnd.com.oracle.cloud.AssemblyInstance+json]	37
185	9.9 Scalability Group [application/vnd.com.oracle.cloud.ScalabilityGroup+json]	40
186	9.10 Server [application/vnd.com.oracle.cloud.Server+json]	43
187	9.11 VM [application/vnd.com.oracle.cloud.VM+json]	47
188	9.12 Volume [application/vnd.com.oracle.cloud.Volume+json]	47
189	9.13 Archive [application/vnd.com.oracle.cloud.Archive+json]	49
190	9.14 Snapshot [application/vnd.com.oracle.cloud.Snapshot+json]	51
191	9.15 VNet [application/vnd.com.oracle.cloud.VNet+json]	52
192	9.16 NetworkInterface [application/vnd.com.oracle.cloud.NetworkInterface+json]	53
193	10 Requests to the Cloud	55
194	10.1 Get Cloud	55
195	11 Operations on VDC resources	56
196	11.1 Get VDC	57
197	11.2 Creates a new VDC	58
198	12 Operations on Zone resources	60
199	12.1 Get Zone	60
200	13 Operations on VM resource	61
201	13.1 Get VM	61
202	13.2 Resize VM	62
203	13.3 Control VM status	64

204	13.4	Add or Remove Network interfaces	65
205	13.5	Adding or Removing Volume	67
206	13.6	Creating a VM Archive	67
207	14	Operations on Volume resources	69
208	14.1	Get Volume	69
209	14.2	Create a new Volume	70
210	14.3	Resize Volume	72
211	14.4	Creating a Volume Snapshot	73
212	15	Operations on Archive resources	74
213	15.1	Get Archive	74
214	15.2	Rolling back a resource to an archive	75
215	16	Operations on VNet resources	77
216	16.1	Get VNet	77
217	16.2	Create a VNet	78
218	16.3	Delete a VNet	79
219	17	Operations on NetworkInterface resources	81
220	17.1	Get Network Interface	81
221	17.2	Create Network Interface	82
222	18	Operations on ScalabilityGroup resources	83
223	18.1	Get ScalabilityGroup	83
224	18.2	Creates a new ScalabilityGroup	84
225	18.3	Shutdown the ScalabilityGroup	85
226	18.4	Scale the ScalabilityGroup	87
227	19	Operations on the ServiceTemplate resource	88
228	19.1	Retrieve information about the service template	88
229	19.2	Deletes a service template	89
230	20	Operations on AssemblyInstance resources	89
231	20.1	Get Assembly Instance	90
232	20.2	Creates a new Assembly Instance	91
233	20.3	Shutdown the Assembly Instance	92
234		Bibliography	Error! Bookmark not defined.
235			

236 Tables

237	Table: Request Headers	16
238	Table: Request Parameters	16
239	Table: Response Headers	18
240	Table: HTTP Status Codes	18
241	Table: ResourceState data model	20
242	Table: Collection<*> data model	21
243	Table: Messages data model	22
244	Table: Message data model	22
245	Table: Common Message Namespace	23
246	Figure: Oracle Cloud Resource Models	25
247	Table: Cloud data model	25
248	Table: ServiceTemplate data model	27
249	Table: VMTemplmate data model	30
250	Table: AssemblyTemplmate data model	32
251	Table: Zone data model	33

Oracle Cloud Management API

252	Table: VDC data model.....	35
253	Table: AssemblyInstance data model.....	37
254	Table: ScalabilityGroup data model.....	40
255	Table: Server data model.....	43
256	Table: Volume data model.....	47
257	Table: Archive data model.....	50
258	Table: Snapshot data model.....	51
259	Table: VNet data model.....	52
260	Table: NetworkInterface data model.....	54
261		

262

Foreword

263 This document introduces the specification of the Cloud Resource Model in the IaaS space and includes
264 some examples using the protocol as described.

265 **Acknowledgements**

266 The authors wish to acknowledge the following people.

267 **Editors:**

268 **Jack Yu**

269 **Mark Carlson**

270 **Contributors:**

271 Sudip Datta

272 Tony Dicenzo

273 Madhup Gulati

274 Kurt Hackel

275 Guanghai Ho

276 Eunhei Jang

277 Sushil Kumar

278 Vadim Lander

279 Ashok Malhotra

280 Adam Messinger

281 Prateek Mishra

282 Gil Pilz

283 John Stanford

284 Garret Swart

285 William Vambenepe

286

287

Oracle Cloud Management Model

288 1 Scope

289 This clause describes the scope of this specification, including some items that are specifically out of
290 scope.

291 1.1 In-Scope

292 Resource Models

293 Examples of Resource Models usages in the REST protocol

294 Extensions to HTTP status code and overloading of existing status code in the context of the use cases

295 1.2 Out of Scope

296 Detailed Security Mechanisms

297 Privilege and Identity Management Model

298 User Models

299 1.3 Conformance

300

301 2 Normative References

302 The following reference documents are indispensable for the application of this document. For dated
303 references, only the edition cited applies. For undated references, the latest edition of the referenced
304 document (including any amendments) applies.

305 Hypertext Transfer Protocol HTTP/1.1 - [RFC 2616](#)

306 HTTP Authentication - [RFC 2617](#)

307 Key words for use in RFCs to Indicate Requirement Levels - [RFC 2119](#)

308 The application/json Media Type for JavaScript Object Notation (JSON) - [RFC4627](#)

309 Media Type Specifications and Registration Procedures - [RFC4288](#)

310

311 3 Terms and Definitions

312 3.1

313 **can**

314 used for statements of possibility and capability, whether material, physical, or causal

- 315 **3.2**
316 **cannot**
317 used for statements of possibility and capability, whether material, physical or causal
- 318 **3.3**
319 **conditional**
320 indicates requirements to be followed strictly in order to conform to the document when the specified
321 conditions are met
- 322 **3.4**
323 **mandatory**
324 indicates requirements to be followed strictly in order to conform to the document and from which no
325 deviation is permitted
- 326 **3.5**
327 **may**
328 indicates a course of action permissible within the limits of the document
- 329 **3.6**
330 **need not**
331 indicates a course of action permissible within the limits of the document
- 332 **3.7**
333 **optional**
334 indicates a course of action permissible within the limits of the document
- 335 **3.8**
336 **shall**
337 indicates requirements to be followed strictly in order to conform to the document and from which no
338 deviation is permitted
- 339 **3.9**
340 **shall not**
341 indicates requirements to be followed strictly in order to conform to the document and from which no
342 deviation is permitted
- 343 **3.10**
344 **should**
345 indicates that among several possibilities, one is recommended as particularly suitable, without mentioning
346 or excluding others, or that a certain course of action is preferred but not necessarily required
- 347 **3.11**
348 **should not**
349 indicates that a certain possibility or course of action is deprecated but not prohibited
- 350 **3.12**
351 **unspecified**
352 indicates that this profile does not define any constraints for the referenced CIM element or operation

353 4 Symbols and Abbreviated Terms

354

355 5 Executive Summary

356

357 The Oracle Cloud API defines an Application Programming Interface (API) to consumers of IaaS
358 clouds based on Oracle's solution stack.

359

360 **Cloud computing** is a style of computing in which dynamically scalable and deployed resources are
361 provided as a service over the network. Users need not have knowledge of, expertise in, or control
362 over the underlying infrastructure in the "cloud" that supports the services rendered to the users. As
363 enterprises (companies, governments, and other organizations) integrate their existing IT
364 infrastructures and IT resources with the sharable cloud paradigm, it is imperative for cloud enablers
365 to provide a uniform API that these enterprises can use to tailor the cloud to their business processes
366 and economic models.

367

368 As IT deployments becoming more complex, an abstraction of the infrastructure resources become
369 more relevant to address concerns of compliance and configuration. Furthermore, such abstractions
370 enable consumers to both self serve, "ala carte" the exact service they need, and to operationally
371 control these services without any significant administrator involvement.

372

373 This API enables an infrastructure provider to service their customers by allowing them to

374

- Browse templates that contain definitions and metadata of a logical unit of service
- Deploy a template into the cloud and form an IT topology on demand
- Perform operations (such as ONLINE, OFFLINE) on the resources
- Take backups of the resources

375

376

377

378

379 The RESTful (**R**epresentational **S**tate **T**ransfer) API presented here focuses on the resource models
380 and their attributes.

381

382 6 Introduction

383 Usage of the API is via the HTTP protocol. The GET, POST, PUT, and DELETE requests are all used.
384 Resource representations documented here are in JSON.

385

386 The API presupposes no particular structure in the URI space. The starting point is a URI, supplied by the
387 cloud service provider, which identifies the cloud itself. The cloud's representation contains URIs for the
388 other resources in the cloud. Operations on the cloud resources are performed by making an HTTP
389 request against the URI of the resource.

390

391 The specification of this Cloud API includes:

392

- Common behaviors that apply across all requests and responses, error messages, common resource attributes
- Resource models, which describe the JSON data structures used in requests and responses
- The requests that may be sent to cloud resources, and the responses expected. Common behaviors would not be described for each resource

393

394

395

396

397

398

399

400 7 Specification

401 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD
402 NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document, and all other documents that
403 comprise the specification of The Platform RESTful API, are to be interpreted as described in “Key
404 words for use in RFCs to Indicate Requirement Levels” ([RFC 2119](#)).

405 The version of this specification and the document version are indicated at the title page.

406 The following are the sections in this specification

- 407 ▪ [Common Behaviors](#)
- 408 ▪ [Resource Models](#)
- 409 ▪ Requests and Responses to
 - 410 ○ [Cloud](#)
 - 411 ○ [VDC](#)
 - 412 ○ [Zone](#)
 - 413 ○ [VM](#)
 - 414 ○ [Volume](#)
 - 415 ○ [Archive](#)
 - 416 ○ [VNet](#)
 - 417 ○ [NetworkInterface](#)
 - 418 ○ [ServiceTemplate](#)
 - 419 ○ [AssemblyInstance](#)
 - 420 ○ [ScalabilityGroup](#)

421

422

423

424 **8 Common Behaviors**

425 This document specifies constraints that apply to all the requests and responses that occur in the RESTful
426 APIs supported by the Oracle Cloud Computing Platform, hereinafter referred to as “The Platform”.

427

428 **8.1 Transport Protocol**

429 All of The Platform APIs are based on the Hypertext Transfer Protocol, version 1.1 ([RFC 2616](#)). Each
430 request will be authenticated using HTTP Basic Authentication ([RFC 2617](#)) unless otherwise noted.
431 Therefore, requests sent from clients on the public Internet (and not on a secure channel such as a VPN)
432 MUST use the https protocol. TLS 1.1 shall be implemented by the provider and TLS 1.2 is strongly
433 encouraged. When TLS is implemented, the following cipher suites shall be supported to ensure a
434 minimum level of security and interoperability between implementations:

- 435 ▪ TLS_RSA_WITH_AES_128_CBC_SHA (mandatory for TLS 1.1/1.2)
- 436 ▪ TLS_RSA_WITH_AES_256_CBC_SHA256 (addresses 112-bit security strength
437 requirements)
- 438 ▪ TLS_RSA_WITH_NULL_SHA (for TLS without encryption)

439 **8.2 URI Space**

440 The resources in the system are identified by URIs. To begin operations, a client must know the URI for a
441 resource. Dereferencing the URI will yield a representation of the resource containing resource attributes
442 and links to associated resources.

443

444 Clients MUST NOT make assumptions about the layout of the URIs or the structures of the URIs of the
445 resources.

446 **8.3 Media Types**

447 In this specification, resource representations and request bodies are encoded in JSON, as specified in
448 [RFC4627](#)

449

450 Each type of resource has its own media-type, which matches the pattern
451 application/vnd.com.oracle.cloud.Xxxxx+json, where “Xxxxx” represents the portion of the identifier unique
452 to a particular representation format for each resource. The identifier MUST be globally unique in the
453 space of vnd.com.oracle.cloud, and the media type should be registered in accordance to [RFC4288](#).

454

455 The Platform MUST provide representations of all resources available in JSON.

456

457 The Platform MUST accept requests from clients encoded in JSON.

458 **8.4 Request Headers**

459 In requests made to services implementing Oracle Cloud Platform APIs, several specific HTTP headers are
 460 used as described in the following table:

461

462

Table: Request Headers

0	Header	Supported Values	Description	Required
1	Accept	Comma-delimited list of media types or media type patterns	Indicates to the server what media type(s) this client is prepared to accept	Recommended, on requests that will produce a response message body
2	Authorization	“Basic “ plus username and password (per RFC 2617)	Identifies the user making this request	Yes on most of the requests
3	Content-Length	Length (in bytes) of the request message body	Describes the size of the message body	Yes, on requests that contain a message body
4	Content-Type	Media type describing the request message body	Describes the representation and syntax of the request message body	Yes, on requests that contain a message body
5	Host	Identifies the host receiving the message	Required to allow support of multiple origin hosts at a single IP address	All requests
6	X-YYYYY-Client-Specification-Version	String containing a specification version number	Declares the specification version of the YYYYY API that this client was programmed against	No

463

464 **8.5 Request Parameters**

465 The client can use request parameters in requests to formulate the following

466

467

Table: Request Parameters

0	Format	Description	Example
---	--------	-------------	---------

1	?attr1,attr2,...	<p>Comma separated attribute names to return the specified attributes of a resource.</p> <p>If an attribute is not part of the resource, then it would be ignored.</p> <p>If none of the attributes is part of the resource, then the resource would be returned in its complete form</p>	<p>Server132?name,description ,status</p> <p>Would return only "name", "description", "status" attributes of the Server132.</p>
2	?[attr_regex]	<p>Attribute regular expression.</p> <p>If none of the resource attributes match the pattern, then the resource would be returned in its complete form.</p> <p><resource_uri> is equivalence to <resource_uri>?[*]</p>	<p>Server132?[.*contain.*]</p> <p>Would return contained_in and container_type attributes of the Server132</p>
3	?[collapse]	<p>This would collapse all the Collection attributes by not returning the individual elements</p>	<p>Only the Collection's uri, name, and total would be returned for all the attributes that are of Collection type</p>
4	?<attr1>:[collapse]	<p>This would return only attr1, and if attr1 is a Collection, it would be collapsed. If attr1 is not a Collection, it would be ignored</p>	<p>?servers:[collapse]</p> <p>Would return Collection's uri, name, and total only</p>
5	?[verbose]	<p>This would show all the fields of all the attributes, recursively, including the collections</p>	<p>For example, /server123?[verbose] would return the expanded list of all the volumes, VNet interfaces, and snapshots, including all the attributes</p>

468 The client must URL encode the request parameters.

469 8.6 Response Headers

470 In responses returned by The Platform, several specific HTTP headers are used as described in the
471 following table:

472

473

Table: Response Headers

0	Header	Supported Values	Description	Required
1	Content-Length	Length (in bytes) of the response message body	Describes the size of the message body	Yes, on responses that contain a message body
2	Content-Type	Media type describing the response message body	Describes the representation and syntax of the response message body	Yes, on responses that contain a message body
3	Location	Canonical URI of the resource, either newly created, or the original referenced resource	Returns a URI that can be used to request a representation of the resource	Yes, on responses to requests that create a new resource, or change an existing resource.
4	Cache-Control	Max-age, public, no-store, must-revalidate, proxy-revalidate	How the representation of the resource should be cached, and its freshness	No. For public resources (such as list of public assemblies or templates) that do not change frequently, allowing lenient cache-control would optimize the response This will never be returned on a privileged resource or a resource request that contains authorization header

474

475 8.7 HTTP Status Codes

476 Oracle Cloud Computing Platform APIs will return standard HTTP response codes as described in the
477 following table, under the conditions listed in the description.

478

479

Table: HTTP Status Codes

0	Status	Description
1	100 Continue	The client SHOULD continue with its request. This interim response is used to inform the client that the initial part of the

		request has been received and has not yet been rejected by the platform. The client SHOULD continue by sending the remainder of the request or, if the request has already been completed, ignore this response.
2	200 OK	The request was successfully completed. If this request created a new resource that is addressable with a URI, and a response body is returned containing a representation of the new resource, a 200 status will be returned with a Location header containing the canonical URI for the newly created resource
3	201 Created	A request that created a new resource was completed, and no response body containing a representation of the new resource is being returned. A Location header containing the canonical URI for the newly created resource will be returned
4	202 Accepted	The request has been accepted for processing, but the processing has not been completed. Per the HTTP/1.1 specification, the returned entity (if any) SHOULD include an indication of the request's current status. A Location header containing the canonical URI for the not-yet completed resource would be returned along with the Status attribute indicating its progress
5	400 Bad Request	The request could not be processed because it contains missing or invalid information (such as validation error on an input field, a missing required value, and so on)
6	401 Unauthorized	The authentication credentials included with this request are missing or invalid
7	403 Forbidden	The server recognized your credentials, but you do not possess authorization to perform this request
8	404 Not Found	The request specified a URI of a resource that does not exist
9	405 Method Not Allowed	The HTTP verb specified in the request (DELETE, GET, HEAD, POST, PUT) is not supported for this request URI
10	406 Not Acceptable	The resource identified by this request is not capable of generating a representation corresponding to one of the media types in the Accept header of the request
11	409 Conflict	A creation or update request could not be completed, because it would cause a conflict in the current state of the resources supported by the platform (for example, an attempt to create a new resource with a unique identifier already assigned to some existing resource or an attempt to modify a resource attribute which is not yet completed)
12	410 Gone	The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent. Clients with link editing capabilities SHOULD delete references to the Request-URI after user approval. If the server does not know, or has no facility to determine, whether or not the condition is permanent,

		the status code 404 (Not Found) SHOULD be used instead. This response is cacheable unless indicated otherwise
13	412 Precondition Failed	The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server. This response code allows the client to place preconditions on the current resource meta-information (header field data) and thus prevent the requested method from being applied to a resource other than the one intended
14	500 Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request
15	501 Not Implemented	The server does not (currently) support the functionality required to fulfill the request
16	503 Service Unavailable	The server is currently unable to handle the request due to temporary overloading or maintenance of the server
The following are extensions		
E1	421 Dependency Not Allowed	The request would result in a broken dependency for associated resources

480

481 **8.8 Common Resource Attributes**

482 All the resource entities in this specification may contain the following common resource attributes.

483 **8.8.1 ResourceState**

484 This attribute denotes the state of the resource describing the lifecycle of the resource. This differs from
 485 the status of the entity represented by the resource which has entity specific semantics.

486

487 The following table is the data model of this attribute.

488

489

Table: ResourceState data model

0	field	Type	Occurs	Description
1	state	String	1	Current state of the resource as last known. This is a label containing lifecycle state (e.g. INITIATED, CREATING, CREATED, DESTROYING, DESTROYED, READY). When there are vendor extensions, the vendor SHALL publish and document their semantics

2	progress	Integer between 0 and 100	0..1	Indicates the progress made as an approximate percentage. Not all state labels assign semantic meaning to this field
3	messages	Message[]	0..1	Include the message data model instances to denote noteworthy communications

490

491 8.8.2 Collection [application/vnd.com.oracle.cloud.Collection+json]

492 This attribute is a meta resource that represents a collection field in a resource. For example, a VDC
 493 contains a collection of VMs, and the field that represents the list of VMs would be implemented in this
 494 type.

495

496 In the resource model, a collection field would be denoted as Collection<type>, for example,
 497 Collection<VM>.

498

499

Table: Collection<*> data model

0	field	Type	Occurs	Description
1	uri	URI	1	The URI that represents the collection of entities
2	type	String	1	The type of the entity that this collection contains
3	total	Integer	0..1	The total number of elements that can be safely assumed to be in the elements list
4	elements	<TYPE>[]	0..1	The list of entities in this collection. At least the uri of the entities must be populated by the platform. When dereferencing the uri, the client must use the type field in the Accept header (except in the case where type = "URI") If this is not returned, then the collection is an empty list

500

501 In addition to the resource type, the collection also supports Collection<URI> where the *type* field is "URI".
 502 This basic type collection would require additional type casting where the uri can be dereferenced properly.

503

504 It is also permissible to have a collection of type collection, for example, Collection<Collection<Server>>.

505

506 **8.9 Error Response Message**

507 **[application/vnd.com.oracle.cloud.common.Messages+json]**

508 Successful requests will generally return an HTTP status code of 200 (OK), 201 (Created), 202 (Accepted),
 509 or 204 (No Content), to indicate that the requested action has been successfully performed or submitted. In
 510 addition, they might include a response message body (with an appropriate media type) containing a
 511 representation of the requested information. However, it is possible for a number of things to go wrong. The
 512 various underlying causes are described (as discussed in the previous section) by various HTTP status
 513 codes in the range 400-499 (for client side errors) or 500-599 (for server side problems).

514

515 If a response is returned with an error status code (400-499 or 500-599), the server SHALL also return a
 516 response message body containing a messages data model, containing zero or more message data
 517 models, describing what went wrong. The text values of such messages might be used, for example, to
 518 communicate with a human user of the client side application.

519

520 The entire list of messages included in a single error response is encapsulated in a messages data model.
 521 The media type SHALL be returned in the Content-Type header. The client SHALL NOT include the
 522 Messages media type in the Accept header.

523

524

Table: Messages data model

0	field	Type	Occurs	Description
1	message	Message	0..n	Zero or more message data for each individual message.

525

526 An individual message contains the following fields:

527

Table: Message data model

0	field	Type	Occurs	Description
1	code	String	0..1	Symbolic error code identifying the type of error reported by this message
2	field	String	0..1	Name of the field from the request data model that this message is associated with
3	hint	String	0..1	Localized text further describing the nature of the problem, possibly including potential workarounds that the client could try

4	text	String	1	Localized text describing the nature of the problem reported by this message
5	severity	String	0..1	Label indicating the severity of the error condition represented by this message Vendor SHALL publish the enumerators that are associated with this field and their semantics
6	stack_trace	String	0..1	Vendor specific stack trace associated with this message
7	source	String	0..1	Symbolic identifier of the service implementation component that triggered this message
8	uri	URI	1	A unique URI that reference this particular message
9	namespace	URI	0..1	A reference to the standard URI to indicate the meaning of this message

528 The *namespace* attribute indicates the semantic meaning of the message which clients may handle
 529 automatically. Messages with the same namespace MUST adhere to the semantic requirement of that
 530 namespace, but the payload (hint, text, severity, stack_trace) may be different. In other words, given a
 531 namespace, clients processing the message should be able to subsequently interact with the providers in a
 532 consistent manner across.

533 Each provider MAY extend the namespace to include specific scenarios and use cases.

534 The information captured in the messages data element SHOULD be complementary to the HTTP status
 535 code, and COULD provide more detailed information. However, it MUST NOT contradict the HTTP status
 536 code that is returned with the request.

537 The following table outlines the common namespace that would accompany this specification

538

Table: Common Message Namespace

0	Namespace	Description
1	/msg/unknown	Unknown error and information given is descriptive in nature
2	/msg/security	Security issues
3	/msg/security/authentication	An authentication error
4	/msg/access	Access violation error
5	/msg/allocation	Allocation related issues
6	/msg/allocation/insufficient	Insufficient resource to satisfy the request

7	/msg/infrastructure	Infrastructure related issues
8	/msg/infrastructure/maintenance	The request cannot be immediately responded due to the infrastructure being in maintenance status

539

540 8.10 Extensibility to the resource model

541 To support returning selective attributes, the following MUST be observed by the service provider:

- 542 ▪ The attribute name of a resource MUST contain alphanumeric characters with “_” and “-“. Thus, [a-
543 zA-Z0-9_\-]

544 9 Cloud Resource Models

545 This section specifies the representations of the resources which this API operates on. The representations
546 are made up of fields, each with a name and value, encoded using a JSON dictionary. The values may be
547 numeric or string literals, lists, or dictionaries, each of which is represented in the obvious way in JSON in
548 accordance to [RFC 4627](#). Clients SHALL NOT assume the order of the fields returned in a response. The
549 number in the table indicates the row number of the table, NOT the order of the fields.

550

551 Each type of cloud resource has its own Internet Media Type. The media type SHALL conform to the
552 pattern application/vnd.com.oracle.cloud.Xxxxxxxx+json, and the specific media type for each resource
553 model is included in square brackets in the corresponding section header.

554

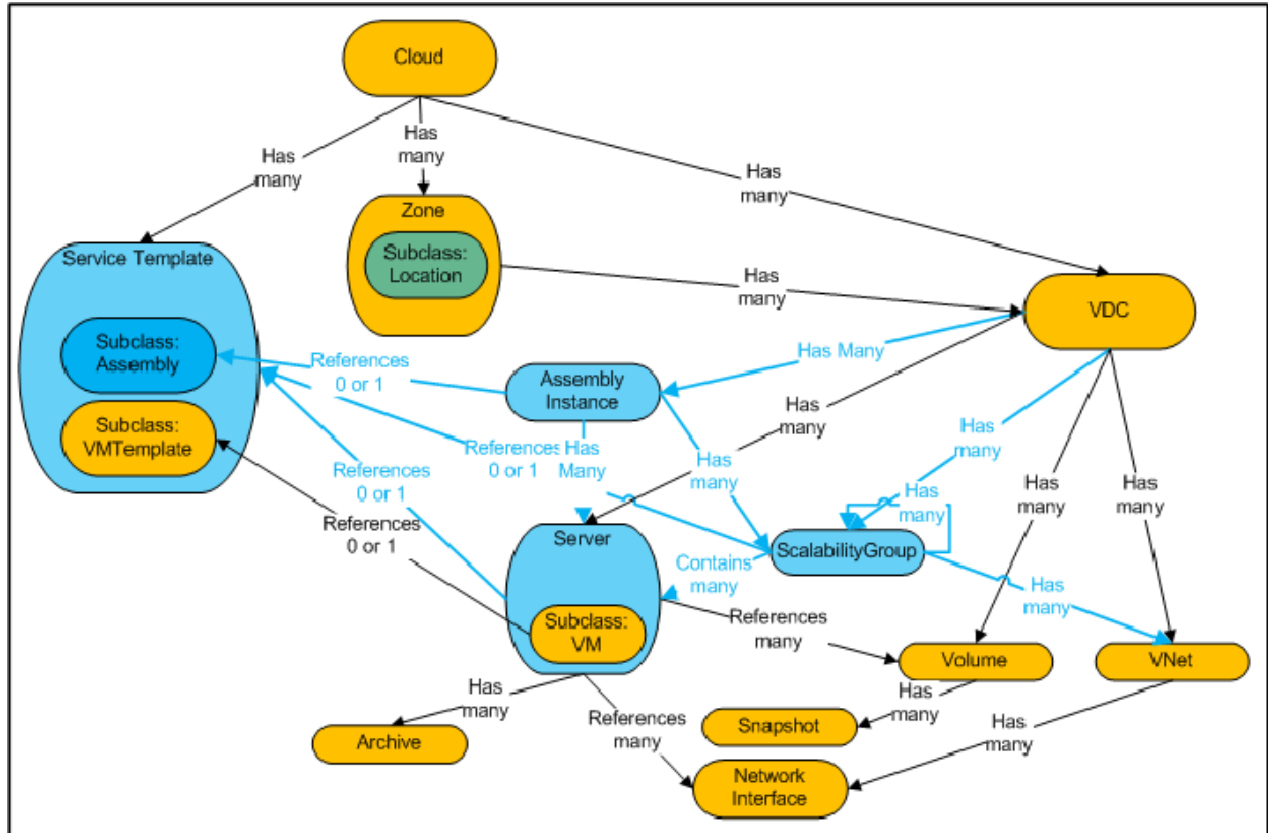
555 In the resource model descriptions, fields annotated with [POST] may be included in a POST request,
556 which is normally used to create new resources. Likewise, fields annotated with [PUT] may be included in a
557 PUT request, which is normally used to update properties of existing resources. Fields not so annotated
558 SHOULD NOT be included in the request body of PUT and POST requests, and SHALL be ignored by the
559 platform if they are included. For a Collection field annotated with [POST-c], the URI of the collection can
560 be posted to add entities into the collection.

561 This specification extends the elemental resources to support resources that are composite in nature, and
562 introduces resources where common attributes are expressed. These resources combined with the
563 elemental resource models form the bases for the Oracle Cloud resource model. In other words, the
564 Oracle Cloud resource model can be “shrink-to-fit” the elemental.

565 The following figure illustrates the view of the resource models defined in this document.

566

Figure: Oracle Cloud Resource Models



567

568

569 **9.1 Cloud [application/vnd.com.oracle.cloud.Cloud+json]**

570 For a user, a Cloud represents the user’s starting view of all accessible resources and deployed entities.

571

572

Table: Cloud data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the resources accessible to this user
2	specification_version	String[]	1	Which version(s) of this specification this server implementation supports
3	implementation_version	String	1	Vendor specific

				version of the server implementation
4	name	String	1	A human readable name. SHALL be a UNICODE string to support different languages
5	description	String	0..1	A brief description. SHALL be a UNICODE string to support different languages
6	tags	String[]	0..1	Values assigned to the cloud by the vendor. Can be used for keywording and terms-of-interests
7	service_templates	Collection <ServiceTemplate>	0..1	The list of service templates that are accessible to the user (Change from VMTemplate)
8	zones	Collection<Zone>	1	List of Zones that are supported by the cloud
9	vdcs	Collection <Vdc>	0..1	Virtual data centers accessible to this user
10	resource_state	ResourceState	0..1	A cloud that is online and running would have READY as its state. If this field is not returned, the client can assume the cloud is READY. If the state of the returned field is not READY, the client cannot assume the viability of subsequent interactions into the cloud
11	profiles	{String,String[][]}	0..1	The list of profiles, each containing groups of service

			<p>characteristics of the cloud, where the vendor may provide to support deployment of resources. The first string indicate the name of the profile, for example, USWest, while the second string list indicates the characteristics of the profile USWest.</p> <p>For example, a service characteristic could be "HIPPA compliance" which is in profile USWest and USEast, but not in EastAsia nor SouthAsia.</p> <p>Each element in the profile list MAY have a zone that can support that profile</p>
--	--	--	--

573

574 **9.2 ServiceTemplate [application/vnd.com.oracle.cloud.ServiceTemplate+json]**

575 For a user, a ServiceTemplate represents the definition of the deployable service. Users can create cloud
 576 resources by specifying the URI of a ServiceTemplate as a field in a deployment request. The cloud
 577 SHALL instantiate the resources and their configurations as specified in the definition of the
 578 ServiceTemplate.

579 Resource, such as VMTemplate, is a subclass to this resource. Thus, it is permissible to get
 580 ServiceTemplate from an uri of a VMTemplate.

581

Table: ServiceTemplate data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the ServiceTemplate definition to this user

2	name	String	1	A human readable name, given to the ServiceTemplate. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	A brief description, given to the ServiceTemplate. SHALL be a UNICODE string to support different languages [POST][PUT]
4	type	String	1	The enumerated String that describes the media type of the service template that the service provide can support [POST]
5	created	Timestamp	1	Date and time, in ISO 8601 format, when the ServiceTemplate was created
6	definition	String	1	The definition of the service template represented in some format, such as XML, that contains all the metadata necessary for the cloud to deploy the service. For example, an ASSEMBLY TEMPLATE's definition could

				contain XML in accordance to the OVF specification [POST][PUT]
7	based_on	Snapshot	0..1	The snapshot of the resource of which this Service Template is based on [POST]
8	tags	String[]	0..1	Free-form values assigned to the Service Template, and can be assigned by the vendor and the clients. Can be used for key-wording or terms-of-interests [POST][PUT]
9	resource_state	ResourceState	1	Only a service template with READY state can be deployed

582

583 The following types of service templates are specified, and they are subclasses of the ServiceTemplate
584 media type

- 585 ▪ VMTemplate – a single OS-stack system on a virtualization platform. When deployed, a
586 Server (in particular, VM) resource would be realized
- 587 ▪ AssemblyTemplate – a system topology that include multiple entities and their
588 interconnections with deployment constraints. The *definition* contains an OVF+
589 Extension. The vendor SHALL publish the appropriate schema to facilitate the parsing
590 of the XML. When deployed, a AssemblyInstance resource would be realized
- 591 ▪ ISOTemplate – an archive file that can be booted to support a Server. When deployed, a
592 Server resource would be realized
- 593 ▪ StorageTemplate – based on the CDMI metadata to specify the characteristics of a
594 volume. When deployed, a Volume resource would be realized
- 595 ▪ NetworkTemplate – the routing relationships and rules that specify the network
596 behaviors. When deployed, a VNet resource would be realized

597 9.3 VMTemplate [application/vnd.com.oracle.cloud.VMTemplate+json]

598 VMTemplate is a class of ServiceTemplate. It is permissible to accept the ServiceTemplate media type on
599 an instance of VMTemplate.

600 VMTemplate is a preconfigured deployable entity that realizes a VM resource.

601

Table: VMTemplate data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the vm template definition to this user
2	name	String	1	A human readable name, given to the VMTemplate. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	A brief description, given to the VMTemplate. SHALL be a UNICODE string to support different languages [POST][PUT]
4	created	Timestamp	1	Date and time, in ISO 8601 format, when the VMTemplate was created
5	params	{}	0..1	Configuration parameters for this VM Template, keyed by parameter name. The list of system defined configuration parameters [POST]
6	based_on	Snapshot	0..1	The snapshot of the resource of

				which this VM Template is based on [POST]
7	tags	String[]	0..1	Free-form values assigned to the VM Template, and can be assigned by the vendor and the clients. Can be used for key-wording or terms-of-interests [POST][PUT]
8	resource_state	ResourceState	1	Only a vm template with READY state can be deployed
9	os	String	0..1	Operating System running on the VM. [POST]
10	cpu	[Number, Number]	0..1	Default count of CPU cores and default CPU core speed in MHz of the VM when provisioned [POST]
11	memory	Integer	0..1	Default main memory size in MB of the VM when provisioned [POST]
12	disks	{String, Integer}[]	0..1	Default list of local disks and their sizes in GB of the VM when provisioned [POST]

604 **9.4 AssemblyTemplate**
 605 **[application/vnd.com.oracle.cloud.AssemblyTemplate+json]**

606 AssemblyTemplate is a class of ServiceTemplate. It is permissible to accept the ServiceTemplate media
 607 type on an instance of AssemblyTemplate.

608 AssemblyTemplate is a deployable entity that realizes a AssemblyInstance resource that may contain
 609 multiple resources that are interconnected.

610

611

Table: AssemblyTemplmate data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the assembly template definition to this user
2	name	String	1	A human readable name, given to the AssemblyTemplate. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	A brief description, given to the AssemblyTemplate. SHALL be a UNICODE string to support different languages [POST][PUT]
4	created	Timestamp	1	Date and time, in ISO 8601 format, when the AssemblyTemplate was created
5	params	{}	0..1	Configuration parameters for this AssemblyTemplate, keyed by parameter name. The list of system defined configuration

				parameters [POST]
6	tags	String[]	0..1	Free-form values assigned to the Assembly Template, and can be assigned by the vendor and the clients. Can be used for key-wording or terms-of-interests [POST][PUT]
7	resource_state	ResourceState	1	Only a vm template with READY state can be deployed
8	definition	String	1	Contains the specification of the Assembly, usually in OVF XML format [POST]

612

613 **9.5 Zone [application/vnd.com.oracle.cloud.Zone+json]**

614

615 A zone represents a logical boundary where the resources may reside. For example, a zone can represent
 616 a particular geographically location such as Europe Zone, North America Zone, East Asia Zone, and so
 617 forth. A zone can also represent characteristics such as high network bandwidth or DMZ secured.
 618 Furthermore, a zone can be organizational in nature, such as Financial Department Zone, Testing Zone,
 619 Development Zone and so forth.

620

621 There should not be any assumption of exclusivity of underlying infrastructures in the zones unless
 622 otherwise noted. For example, Zone A and Zone B can be on the same physical network serving two
 623 different departments, but their physical infrastructure setup is transparent to cloud users.

624

625 The zone SHALL support the union of the service characteristics of the list of the profiles. The relationship
 626 between Zone/Profile/Characteristics is:

- 627 ▪ Profile contains a list of service characteristics
- 628 ▪ Zone is assigned profiles

629

630

Table: Zone data model

0	field	Type	Occurs	Description
---	-------	------	--------	-------------

1	uri	URI	1	A GET against this URI refreshes the client representation of the Zone definition to this user
2	name	String	1	Name of the Zone. SHALL be a UNICODE string to support different languages
3	description	String	0..1	Human readable description of the Zone. SHALL be a UNICODE string to support different languages
4	tags	String[]	0..1	Vendor specific values assigned to the zone by the vendor. Can be used for key-wording or terms-of-interests
5	profiles	String[]	0..1	This field indicates the list of characteristics that this zone supports. This SHALL be the subset of the profiles of the cloud that contains this zone The Zone SHALL support the union of the service characteristics of the profiles
6	platform	String	0..1	This field indicates the underlying platform technology supporting the zone. This is an enumeration of the following values: Zen, Esx, LDom, Solaris, HyperV, Physical If specified, the zone would uniformly support the platform. (Add Physical to the enumerator to indicate non-virtualized topology)

631

632 **9.6 Location [application/vnd.com.oracle.cloud.Location+json]**

633 Location is a resource that is a subclass of the Zone resource
 634 [application/vnd.com.oracle.cloud.Zone+json]. Location has all the fields of the Zone with identical
 635 semantics. Vendors SHALL document, when appropriate, the additional parameters that a Location may
 636 need to contain. For example, a vendor can decide that a Location must contain an entry in the tags field
 637 that starts with "Location=".

638

639 A client SHALL be able to perform a GET request with accept:Zone on a Location resource, since a
 640 Location is also a Zone. However, a vendor SHALL document when it is NOT appropriate for a client to
 641 perform a GET operation with accept:Location on a Zone resource.

642

643 9.7 VDC [application/vnd.com.oracle.cloud.VDC+json]

644 A VDC represents a user's view of the grouping of resources that make up a data center. The vendor MAY
 645 enforce underlying resource limitations on a VDC.

646

647

Table: VDC data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the VDC
2	name	String	1	Name of the VDC. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable description of the VDC given by the user. SHALL be a UNICODE string to support different languages [POST][PUT]
4	tags	String[]	0..1	Values assigned to the VDC by the user. Can be used for key-wording or terms-of-

				interests [POST][PUT]
5	scalability_groups	Collection<ScalabilityGroup>	0..1	A list of scalability groups that are included in this VDC [POST-c]
6	servers	Collection <Server>	0..1	A list of Servers that are directly included in this VDC [POST-c]
7	volumes	Collection<Volume>	0..1	A list of volumes that are directly included in this VDC [POST-c]
8	vnets	Collection<VNet>	0..1	A list of virtual network services that are directly included in this VDC [POST-c]
9	zone	URI	1	The uri of the zone of which the VDC belongs to. All resources in the VDC are presumed to be in the zone [POST]
10	resource_state	ResourceState	1	The validity of the other VDC fields on a GET should be guaranteed only when the resource state is READY.

				Otherwise, the client should not assume the validity of the fields
11	created	Timestamp	1	Date and time, in ISO 8601 format, when the VDC was created
12	params	{ }	0..1	Configuration parameters for this VDC [POST]
13	assembly_instances	Collection <AssemblyInstance>	0..1	Assembly Instances that are deployed within this VDC [POST-c] (Addition)

648

649 9.8 AssemblyInstance

650 [application/vnd.com.oracle.cloud.AssemblyInstance+json]

651 An instance of an AssemblyInstance is a logical grouping of resources from a deployment request of an
652 Assembly template. The lifecycle of the resources in an assembly instance could be managed centrally via
653 the assembly instance.

654

655 **Important note:** a service template of Assembly type would be deployed into an instance of a
656 AssemblyInstance.

657

658

Table: AssemblyInstance data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the Assembly Instance definition

2	name	String	1	Name of the Assembly Instance as given by the user. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable description of the Assembly Instance given by the user. SHALL be a UNICODE string to support different languages [POST][PUT]
4	tags	String[]	0..1	Values assigned to the Assembly Instance by the user. Can be used for key-wording or terms-of-interests [POST][PUT]
5	based_on	URI	0..1	The URI of the service template of which this Assembly Instance is based on [POST]
6	scalability_groups	Collection <ScalabilityGroup>	0..1	A list of scalability groups that are included in this Assembly Instance [POST-c]
7	servers	Collection <Server>	0..1	A list of Servers that are directly included in this Assembly Instance

				[POST-c]
8	status	String	1	Indicate the status of the Assembly Instance. This field contains the semantics that the service provider implements. For example, a service provider may implement an ONLINE status to indicate all the entities, recursively, are in an ONLINE status. Or a service provider may implement an ONLINE status to indicate critical entities are in an ONLINE status [POST][PUT]
9	resource_state	ResourceState	1	The validity of the other AssemblyInstance fields on a GET should be guaranteed only when the resource state is READY. Otherwise, the client should not assume the validity of the fields
10	created	Timestamp	1	Date and time, in ISO 8601 format, when the Assembly Instance was created
11	expiry	Timestamp	0..1	Date and time, in ISO 8601 format, when the Assembly Instance should expire. If not specified, the

				Assembly Instance never expires [PUT]
12	params	{ }	0..1	Vendor specific configuration parameters for this deployment [POST][PUT]
13	contained_in	URI	0..1	URI of the VDC that this Assembly Instance is contained in

659

660

661 **9.9 Scalability Group [application/vnd.com.oracle.cloud.ScalabilityGroup+json]**

662 Scalability Group is a collection of servers and corresponding virtual networks. There are 2 types:

- 663 ▪ Homogenous: contain a collection of homogenous entities. The cloud service provider
- 664 SHOULD enforce the semantics of “sameness”. Operations such as scale_out and
- 665 scale_in of the scalability groups MAY be supported by the provider in the
- 666 homogenous scalability group
- 667 ▪ Heterogeneous: contain a collection of entities that do not have the semantics of
- 668 “sameness”

669

670

Table: ScalabilityGroup data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this uri refreshes the client representation of the scalability group definition
2	name	String	1	Name of the scalability group as given by the user or generated by the platform. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable description of the scalability group given by the user or generated by

				the platform. SHALL be a UNICODE string to support different languages [POST][PUT]
4	nodes	Collection<URI>	1	The list of URI's that represent the entities making up this scalability group [POST-c]
5	type	String	1	The type of the entities that make up the scalability group. This is used for the client to formulate the Accept header of the GET request when dereferencing the URI's [POST]
6	count	Integer	1	The count of the nodes that are in the scalability group [PUT]
7	contained_in	URI	1	URI of the ScalabilityGroup, VDC or the Assembly Instance that this scalability group is contained in [POST]
8	container_type	String	1	Either ScalabilityGroup, VDC or AssemblyInstance that this scalability group is contained in. This is used for the client to formulate the Accept header of the GET request when dereferencing the <i>contained_in</i> field [POST]
9	status	String	1	Indicates the status of the scalability group. This field contains the vendor dependent semantics that the service provider implements. For

				<p>example, a service provider may implement an ONLINE status to indicate all the entities, recursively, are in an ONLINE status. Or a service provider may implement an ONLINE status to indicate at least 1 entity is in an ONLINE status</p> <p>[PUT]</p>
10	tags	String[]	0..1	<p>Values assigned to the scalability group by the user or generated by the platform. Can be used for key-wording or terms-of-interests</p> <p>[POST][PUT]</p>
11	resource_state	ResourceState	1	<p>The validity of the other ScalabilityGroup fields is guaranteed only when the resource state is READY. Otherwise, the client SHALL not assume the validity of the fields</p>
12	created	Timestamp	1	<p>Date and time, in ISO 8601 format, when the scalability group is created</p>
13	max	Signed Integer	0..1	<p>The maximum number of nodes this scalability group can hold. If not provided, the client should assume it is unlimited, which is the specified with as the value "-1"</p> <p>[POST][PUT]</p>
14	min	Integer	0..1	<p>The minimal number of nodes this scalability group should hold to be considered a functional scalability group. If not specified, the client should assume it is 1</p> <p>[POST][PUT]</p>
15	vnets	Collection<VNet>	0..1	<p>A list of virtual network services (such as firewall,</p>

				load balancer, vswitch) that are included in this scalability group [POST-c]
16	based_on	URI	0..1	URI of the service template where the homogenous nodes of the scalability group are from [POST]
17	homogenous	BOOLEAN	1	TRUE when the scalability group contains homogenous entities and FALSE otherwise [POST] (Addition)

671

672 **9.10 Server [application/vnd.com.oracle.cloud.Server+json]**

673

674 A Server is a computing container providing a complete system platform that supports the execution of a
675 complete OS stack. On a virtualization platform, a Server is commonly known as a Virtual Machine (VM).

676

677

Table: Server data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the Server definition
2	name	String	1	Name of the Server as given by the user or generated by the platform. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable

				<p>description of the Server given by the user or generated by the platform. SHALL be a UNICODE string to support different languages</p> <p>[POST][PUT]</p>
4	tags	String[]	0..1	<p>Values assigned to the Server by the user or generated by the platform. Can be used for key-wording or terms-of-interests</p> <p>[POST][PUT]</p>
5	contained_in	URI	1	<p>URI of the ScalabilityGroup, VDC or the Assembly Instance that this Server is contained in</p> <p>[POST][PUT]</p>
6	container_type	String	1	<p>Either ScalabilityGroup, VDC or AssemblyInstance that this Server is contained in. This is used for the client to formulate the Accept header of the GET request when dereferencing the <i>contained_in</i> field</p> <p>[POST][PUT]</p>
7	status	String	1	<p>Current running status of this Server. The service provider can overwrite the valid values for this field, and may implement status</p>

				operations. More details on this field follows [PUT]
8	based_on	URI	0..1	The URI of the service template on which this Server is based on [POST]
9	hostname	String	0..1	Qualified host name of this Server
10	cpu	[Number,Number]	1	A numeric sizing of the CPU where the first number indicates the counts of the CPU cores and the second number indicates the CPU speed in MHz per core [POST][PUT]
11	memory	Number	1	A numeric sizing of the RAM in MByte [POST][PUT]
12	disks	{String,Number}[]	0..1	The name and size in GB of local disks
13	volumes	Collection <Volume>	0..1	A list of volumes that are attached to this Server [POST-c]
14	interfaces	Collection <NetworkInterface>	1	Network interfaces associated with this Server [POST-c] (this POST is not eligible for Server with type PHYSICAL)

15	params	{ }	0..1	Vendor specific configuration parameters for this Server [POST][PUT]
16	archives	Collection <Archive>	0..1	A list of archives that have been taken of this Server [POST-c]
17	cloned_from	URI	0..1	If this Server was instantiated from an archive, this field would indicate the uri of the original snapshot [POST]
18	resource_state	ResourceState	1	The validity of the other Server fields is guaranteed only when the resource state is READY. Otherwise, the client should not assume the validity of the fields
19	created	Timestamp	1	Date and time, in ISO 8601 format, when the Server was created
20	type	String	1	The type of the server that represents this computational container (PHYSICAL, VIRTUAL)
21	restored_from	URI	0..1	If this server was restored from an existing archive in its archive list, this field will contain the uri of the archive entity

				[PUT]
--	--	--	--	-------

678

679 The *status* field of the Server data model should contain the running status of the Server. It is expected
680 that the service provider implements at least the following valid values

- 681 ▪ STOPPED
- 682 ▪ STOPPING
- 683 ▪ STARTING
- 684 ▪ STARTED
- 685 ▪ SUSPENDED
- 686 ▪ SUSPENDING
- 687 ▪ RESUMING
- 688 ▪ RESTARTING

689 The service provider SHALL implement valid status transition operations and consistency when providing
690 the Server resource to the client. It should be noted that the status of RESTARTING is a transitional status
691 that is expected to end in STARTED status.

692

693 **9.11 VM [application/vnd.com.oracle.cloud.VM+json]**

694 VM is a resource that is a subclass of the Server resource [application/vnd.com.oracle.cloud.Server+json]
695 by fixing *type*=VIRTUAL. VM has all the fields of the Server with identical semantics.

696

697 A client SHALL be able to perform a GET request with *accept*:Server on a VM resource, since a VM is also
698 a Server. However, a client can only be able to perform a GET with *accept*:VM on a Server resource when
699 it is of the VIRTUAL type. If a Server resource is not of VIRTUAL type, a GET operation with *accept*:VM on
700 a Server resource SHALL return status code 406.

701

702 **9.12 Volume [application/vnd.com.oracle.cloud.Volume+json]**

703

704 An instance of a Volume is a storage entity that may be
705 ▪ Implicitly created on behalf of another entity. For example, instantiating a VM may
706 include the creation of a storage unit (for instance, a LUN)
707 ▪ Explicitly created for the purpose of sharing among different entities. For example, a
708 shared storage that multiple servers can mount with

709 If coupled with the CDMI Cloud Storage standard, this API can be used to inter-operably configure storage
710 for use by the computing cloud.

711

712

Table: Volume data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the

				Volume definition
2	name	String	1	Name of the Volume as given by the user or generated by the platform. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable description of the Volume given by the user or generated by the platform . SHALL be a UNICODE string to support different languages [POST][PUT]
4	tags	String[]	0..1	Values assigned to the Volume by the user or generated by the platform. Can be used for key-wording or terms-of-interests [POST][PUT]
5	contained_in	URI	1	URI of the ScalabilityGroup or the VDC that this Volume is contained in [POST][PUT]
6	container_type	String	1	Either ScalabilityGroup or VDC that this Volume is contained in. This is used for the client to formulate the Accept header of the GET request when dereferencing the <i>contained_in</i> field [POST][PUT]
7	params	{ }	0..1	CDMI ObjectID (if supported) or vendor dependent configuration parameters for this volume. The parameters metadata may be included in the service template, in the CDMI Container specified, and/or provided by the service provider. This includes the quality of

				service characteristics [POST][PUT]
8	size	Number	1	The size of the volume in GBytes [POST][PUT]
9	snapshots	Collection <Snapshot>	0..1	A list of snapshots that have been taken on this volume [POST-c]
10	cloned_from	URI	0..1	If this volume was instantiated from a snapshot, this field would indicate the uri of the original snapshot [POST][PUT]
11	created	Timestamp	1	Date and time, in ISO 8601 format, when the Volume was created
12	resource_state	ResourceState	1	The validity of other Volume fields is guaranteed only when the resource state is READY. Otherwise, the client should not assume the validity of the fields
13	restored_from	URI	0..1	If this volume was restored from an existing snapshot in its snapshot list, this field will contain the uri of snapshot entity [PUT]
14	based_on	URI	0..1	The URI of the StorageTemplate of which this volume is based [POST] (Addition)

713

714 **9.13 Archive [application/vnd.com.oracle.cloud.Archive+json]**

715

716 An archive represents a point-in-time representation of a Server.

717

718

Table: Archive data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the Archive definition
2	name	String	1	Name of the Archive as given by the user or generated by the platform . SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable description of the Archive given by the user or generated by the platform . SHALL be a UNICODE string to support different languages [POST][PUT]
4	created	Timestamp	1	Date and time, in ISO 8601 format, when this archive was created
5	source	URI	1	The uri of the Server from which this archive was taken [POST]
6	resource_state	ResourceState	1	The validity of the other archive fields is guaranteed only when the resource state is READY. Otherwise, the client should not assume the validity of the fields
7	params	{ }	0..1	Configuration parameters for this archive [POST]

719

720

721 **9.14 Snapshot [application/vnd.com.oracle.cloud.Snapshot+json]**

722

723 A snapshot represents a point-in-time representation of a volume.

724

725

Table: Snapshot data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the Snapshot definition
2	name	String	1	Name of the Snapshot as given by the user or generated by the platform . SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable description of the Snapshot given by the user or generated by the platform . SHALL be a UNICODE string to support different languages [POST][PUT]
4	created	Timestamp	1	Date and time, in ISO 8601 format, when this snapshot was created
5	source	URI	1	The uri of the volue from which this snapshot was taken [POST]
6	resource_state	ResourceState	1	The validity of the other Snapshot fields is guaranteed only when the resource state is READY. Otherwise, the client should not assume the validity of the fields
7	params	{ }	0..1	Configuration parameters for this Snapshot [POST]

726

727 **9.15 VNet [application/vnd.com.oracle.cloud.VNet+json]**

728

729 A VNet is a service that is capable of providing network addresses, routing rules, security constraints, and
730 access limits.

731 **Note: VNet resource can be extended to specific network services that the cloud infrastructure**
732 **would provide. This specification does not enumerate various network services (e.g. load balancer,**
733 **network fencing, DMZ, and others) nor does it assume any particular network topologies. The**
734 **future version(s) of this specification may incorporate emerging refinements to this resource.**

735

Table: VNet data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the VNet definition
2	name	String	1	Name of the VNet as given by the user or generated by the platform. SHALL be a UNICODE string to support different languages [POST][PUT]
3	description	String	0..1	Human readable description of the VNet given by the user or generated by the platform. SHALL be a UNICODE string to support different languages [POST][PUT]
4	tags	String[]	0..1	Values assigned to the VNet by the user or generated by the platform. Can be used for key-wording or terms-of-interests [POST][PUT]
5	contained_in	URI	1	URI of the ScalabilityGroup or the VDC that this VNet is contained in [POST]
6	container_type	String	1	The type of the container that this VNet is contained

				in. This is used for the client to formulate the Accept header of the GET request when dereferencing the <i>contained_in</i> field [POST]
7	params	{ }	0..1	Vendor dependent configuration parameters for this VNet [POST][PUT]
8	created	Timestamp	1	Date and time, in ISO 8601 format, when the VNet was created
9	interfaces	Collection <NetworkInterface>	0..1	The list of NetworkInterface resources that are part of the VNet [POST-c]
10	based_on	URI	0..1	The uri of the NetworkTemplate on which this VNet is based [POST] (Addition)
11	base_network	URI	0..1	This uri is the canonical reference of this VNet to a base VNet that may be common across cloud. If two VNets have different base_network references, then the VNets would not be communicatable. If two VNets have the same base_network references, then they MAY be communicatable. If two VNets do not have base_network specified, then they would not be communicatable

736

737 **9.16 NetworkInterface [application/vnd.com.oracle.cloud.NetworkInterface+json]**

738

739 An instance of the network interface is identified by a network end point and consists of a complete address
 740 that can be interpreted by the underlying network infrastructure.

741

742

Table: NetworkInterface data model

0	field	Type	Occurs	Description
1	uri	URI	1	A GET against this URI refreshes the client representation of the NetworkInterface definition
2	name	String	1	Name of the NetworkInterface as given by the user or generated by the platform. SHALL be a UNICODE string to support different languages [PUT]
3	description	String	0..1	Human readable description of the NetworkInterface given by the user or generated by the platform. SHALL be a UNICODE string to support different languages [PUT]
4	routable	Boolean	1	If FALSE, this network interface may not be assumed to be pingable from outside of its immediate boundary. The boundary would be defined and specified by the provider's implementation of the Network of which this network interface comes from. For example, a provider may define the boundary on the Assembly Instance level. In this case, FALSE would mean that the network interface is not accessible by entities that are not in the same Assembly Instance. TRUE would mean that the network interface is accessible by entities that are not in the same Assembly Instance

5	vnet	URI	1	The associated VNet of which this interface is a member of
6	address	String	1	The address of this interface. If an IP based protocol, specified as a dotted notation IPv4 or IPv6 address
7	address_type	String	1	The type of the address for this network interface specified by the vendor. For example, IPv4 or IPv6

743

744

745 10 Requests to the Cloud

746 Requests to cloud allow the client to enumerate resources and entities that the user can access.

747 10.1 Get Cloud

748 Retrieve information about accessible resources to a user.

749

750 **Synopsis:** GET {Well Known URI from the cloud provider}

751 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

752 **Request Message Body:** N/A.

753 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

754 **Response Message Body:** Cloud

755 **Response Status:** 200, 400, 401, 403, 404

756

757 Example Request: retrieving information about accessible resources

```
758 GET /
759 Host: cloudcompany.com
760 Authorization: Basic xxxxxxxxxxxx
761 Accept: application/vnd.com.oracle.cloud.Cloud+json
762 X-Cloud-Client-Specification-Version: 0.1
```

763

764 Example Response: resources that are accessible to the user

```
765 HTTP/1.1 200 OK
766 Content-Type: application/vnd.com.oracle.cloud.Cloud+json
767 Content-Location: https://cloudcompany.com/
768 Cache-Control: no-store
769 Content-Length: nnn
```

```

770 {
771   "uri" : "https://cloudcompany.com/",
772   "specification version" : [ "0.8" ] ,
773   "implementation version" : "3.8RC2",
774   "name" : "Oracle Cloud Service Provider",
775   "description" : "Providing Cloud services to the ABC industry...",
776   "tags" : ["ABC", "Cloud", "Telecom", ... ],/templates
777   "zones" : {
778     "uri" : "/123/zones",
779     "type" : "Zone",
780     "total" : "5" ,
781     "elements" : [
782       {"name" : "USA West Coast",
783        "uri" : "/123/wczone"},
784       {"name" : "Europe West",
785        "uri" : "/123/euzone"},
786       {"name" : "Secured DMZ",
787        "uri" : "/123/sczone"},
788       ... ]
789     },
790   "vdc" : {
791     "uri" : "/123/vdc",
792     "type" : "VDC",
793     "total" : "1" ,
794     "elements" : [
795       {"name" : "Default Work Center",
796        "uri" : "/123/vdc/vdc232"}
797       ... ]
798     },
799   "service templates" : {
800     "uri" : "/templates/items/",
801     "type" : "VMTemplate",
802     "total" : "5" ,
803     "elements" : [
804       {"name" : "Oracle Peoplesoft Sales Demo",
805        "uri" : "/templates/items/t123"},
806       {"name" : "Ruby on Rail with OEL 4",
807        "uri" : "/template/items/t132"},
808       {"name" : "Oracle Database SE 11.2",
809        "uri" : "/template/items/t134"},
810       {"name" : "MySQL Enterprise with OEL5",
811        "uri" : "/templates/items/t839"},
812       {"name" : "Oracle EM NextGen Demo in a box",
813        "uri" : "/templates/items/t833"}
814     ]
815   }
816   "resource state" : {
817     "state" : "READY"
818   },
819   "profiles" : [
820     {"USWest", ["NETWORK LATENCY=LOW", "HA=TRUE"]},
821     {"USEast", ["NETWORK LATENCY=LOW", "HA=TRUE"]},
822     {"US", ["HIPPA=TRUE"]}
823   ]
824 }
825

```

826 [Return to Section List](#)

827 11 Operations on VDC resources

828 Requests to VDC allow the user to

- 829 ▪ View the VDC information, including all of its direct sub components
- 830 ▪ Create a new VDC
- 831 ▪ Add more resources to the VDC
- 832 ▪ Delete a VDC – the associated resources would be cascading removed from the VDC

833 11.1 Get VDC

834 Retrieving the details of a VDC.

835

836 **Synopsis:** GET {URI of a VDC}

837 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

838 **Request Message Body:** N/A.

839 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

840 **Response Message Body:** VDC

841 **Response Status:** 200, 400, 401, 403, 404, 410

842

843 Example Request: Retrieve the VDC attributes

```
844 GET /123/vdcs/vdc232
845 Host: cloudcompany.com
846 Authorization: Basic xxxxxxxxxxxx
847 Accept: application/vnd.com.oracle.cloud.AssemblyInstance+json
848 X-Cloud-Client-Specification-Version: 0.1
```

849

850 Example Response

```
851 HTTP/1.1 200 OK
852 Content-Type: application/vnd.com.oracle.cloud.VDC+json
853 Content-Location: /123/vdcs/vdc232
854 Content-Length: nnn
855 {
856   "uri" : "/123/vdcs/vdc232",
857   "name" : "My Cloud Work Data Center",
858   "description" : "This is the default virtual data center that is
859   created on your behave. This data center is not zone specific" ,
860   "tags" : ["Default", "Data Center"] ,
861   "scalability groups" : {
862     "uri" : "/123/vdcs/vdc232/scalability_groups",
863     "type" : "ScalabilityGroup",
864     "total" : "2" ,
865     "elements" : [
866       { "uri": "/123/vdcs/vdc232/c24", "name": "RAC Database" },
867       { "uri": "/123/vdcs/vdc232/c43", "name": "FMW Application
868 Server - OMS" }
869     ]
870   } ,
871   "servers" : {
872     "uri" : "/123/vdcs/vdc232/servers" ,
873     "type" : "VM" ,
874     "total" : "3" ,
875     "elements" : [
876       { "uri" : "/123/vdcs/vdc232/vm23", "name" : "Website Load
877 Tester" } ,
```

```

878     { "uri" : "/123/vdcs/vdc232/vm34", "name" : "Simple Web
879 Server for serving Reports" },
880     { "uri" : "/123/vdcs/vdc232/vm49", "name" : "Database to
881 store testing results" }
882   ]
883 },
884   "volumes" : {
885     "uri" : "/123/vdcs/vdc232/volumes",
886     "type" : "Volume",
887     "total" : "1",
888     "elements" : [
889       { "uri" : "/123/vdcs/vdc232/vol142", "name" : "Shared
890 Storage for Software Library" }
891     ]
892   },
893   "vnets" : {
894     "uri" : "/123/vdcs/vdc232/networks",
895     "type" : "VNet",
896     "total" : "3",
897     "elements" : [
898       { "uri" : "/123/vdcs/vdc232/net93", "name" : "Load
899 Balancer for OMS Cluster" },
900       { "uri" : "/123/vdcs/vdc232/net103", "name" : "Firewall to
901 protect RAC database" },
902       { "uri" : "/123/vdcs/vdc232/net91", "name" : "Routable
903 Network Profile for Testing EM" }
904     ]
905   },
906   "assembly instances" : {
907     "uri" : "/123/vdcs/vdc232/AssemblyInstances",
908     "type" : "AssemblyInstance",
909     "total" : "1",
910     "elements" : [
911       { "uri" : "/123/dg/34", "name" : "Testing Site: Enterprise
912 Manager with RAC Database 11.2" }
913     ]
914   },
915   "resource state" : {
916     "state" : "READY"
917   },
918   "created" : "February 26, 2010 15:34:34 GMT",
919   "expiry" : "March 26, 2010 15:34:34 GMT"
920 }

```

921

922 11.2 Creates a new VDC

923

924 In this example, the user creates a new VDC in the cloud.

925

926 **Synopsis:** POST {URI of the Cloud or URI of the Cloud's VDCs collection}

927 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 928 Authorization, Accept

929 **Request Message Body:** VDC

930 **Response Headers:** Content-Length, Content-Type, Content-Location

931 **Response Message Body:** VDC

932 **Response Status:** 200, 202, 400, 401, 403, 404, 409

933

934 **Example Request:** creates a zone specific vdc under a cloud

```

935 POST /123/vdcs
936 Host: cloudcompany.com
937 Authorization: Basic xxxxxxxxxxxx
938 Content-Length: nnn
939 Content-Type: application/vnd.com.oracle.cloud.VDC+json
940 Accept: application/vnd.com.oracle.cloud.VDC+json
941 X-Cloud-Client-Specification-Version: 0.1
942 {
943     "name" : "My Data Center in US West Zone" ,
944     "description" : "This is a data center to be encapsulated in the
945     US West Zone" ,
946     "zone" : "/123/wczone" ,
947     "tags" : ["Data Center", "US West"] ,
948     "params" : {
949         "Routable Static IPs" : "TRUE"
950     }
951 }
```

952

953 **Example Response:**

```

954 HTTP/1.1 200 Ok
955 Content-Type: application/vnd.com.oracle.cloud.VDC+json
956 Content-Location: /123/vdcs/103
957 Content-Length: nnn
958 {
959     "uri" : "/123/vdcs/103" ,
960     "name" : "My Data Center in US West Zone" ,
961     "description" : "This is a data center to be encapsulated in the
962     US West Zone" ,
963     "zone" : "/123/wczone" ,
964     "tags" : ["Data Center", "US West"] ,
965     "vnets" : {
966         "uri" : "/123/vdcs/vdc232/networks"
967         "type": "VNet" ,
968         "total" : "1" ,
969         "elements" : {
970             "uri" : "/123/vdcs/vdc232/n234" , "name" : "Static
971     Routable IP Address Container"
972         }
973     }
974     "resource state" : {
975         "state" : "READY" ,
976         "progress" : "100" ,
977         "message" : [
978             { "code" : "2", "text" : "validating" } ,
979             { "code" : "6", "text" : "VDC created" } ,
980             ...
981         ]
982     }
983 }
```

984

985 The VDC created in the "/123/wczone" defaults with one network resource that contains static routable IP
986 addresses. Resources, such as networks, volumes, may be created by service providers based on
987 configurations to be contained in the VDC.

988

989 [Return to Section List](#)

990

991 12 Operations on Zone resources

992 Operations on a Zone resource allow the user to view the attributes of the zone.

993 12.1 Get Zone

994 Retrieve information about the Zone

995

996 **Synopsis:** GET {URI of a Zone}

997 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

998 **Request Message Body:** N/A.

999 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

1000 **Response Message Body:** Zone

1001 **Response Status:** 200, 400, 401, 403, 404

1002

1003 Example Request: Retrieve the Zone attributes

```
1004 GET /123/wczone
1005 Host: cloudcompany.com
1006 Authorization: Basic xxxxxxxxxxxx
1007 Accept: application/vnd.com.oracle.cloud.Zone+json
1008 X-Cloud-Client-Specification-Version: 0.1
```

1009

1010 Example Response

```
1011 HTTP/1.1 200 OK
1012 Content-Type: application/vnd.com.oracle.cloud.Zone+json
1013 Content-Location: //123/wczone
1014 Cache-Control: public
1015 Content-Length: nnn
1016 {
1017   "uri" : "...",
1018   "name" : "USA West Coast",
1019   "description" : "This is the USA West Coast Zone where the
1020 underlying compute, storage, and network resources are in closer
1021 proximity to the west coast of the United State..." ,
1022   "tags" : ["ABC", "USA", "West Coast", ... ] ,
1023   "profiles" : ["USWest", "US"],
1024   "platform" : "Zen"
1025 }
```

1026

1027

1028 [Return to Section List](#)

1029

1030

13 Operations on VM resource

1031 Operation to a VM resource enables the user to

- 1032
 - Retrieve information about a VM
 - 1033
 - Modify the size of the VM
 - 1034
 - Controls the operational status of the VM
 - 1035
 - Add or Remove network interfaces
 - 1036
 - Creates an archive

1037

13.1 Get VM

1038 Retrieving the details of a VM.

1039

1040 **Synopsis:** GET {URI of a VM}1041 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization1042 **Request Message Body:** N/A.1043 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location1044 **Response Message Body:** VM1045 **Response Status:** 200, 400, 401, 403, 404, 410

1046

1047 Example Request: Retrieve the VM attributes

```

1048 GET /123/vdcs/vdc232/vm34
1049 Host: cloudcompany.com
1050 Authorization: Basic xxxxxxxxxxxx
1051 Accept: application/vnd.com.oracle.cloud.VM+json
1052 X-Cloud-Client-Specification-Version: 0.1

```

1053

1054 Example Response

```

1055 HTTP/1.1 200 OK
1056 Content-Type: application/vnd.com.oracle.cloud.Server+json
1057 Content-Location: /123/dg/34/vm34
1058 Cache-Control: no-store
1059 Content-Length: nnn
1060 {
1061   "uri" : "...",
1062   "name" : "Simple Web Server for serving Reports",
1063   "description" : "A single Web Server that can serve the QA
1064 testing reports and ...",
1065   "tags" : ["Testing", "Web Server", "EM", "Reporting", ...] ,
1066   "contained in" : "/123/vdcs/vdc232" ,
1067   "container_type" : "VDC" ,

```

```

1068     "status" : "STARTED" ,
1069     "based on" : "/templates/items/t904" ,
1070     "hostname" : "qareport234.mycompany.com" ,
1071     "cpu" : "2400" ,
1072     "memory" : "2048" ,
1073     "disks" : {
1074         ["/mnt1", "128"]
1075     } ,
1076     "volumes" : {
1077         "uri" : "/123/vdcs/vdc232/vm34/volumes" ,
1078         "type" : "Volume" ,
1079         "total" : "2" ,
1080         "elements" : [
1081             { "uri" : "/123/vdcs/vdc232/vol132", "name" : "VOL 1" } ,
1082             { "uri" : "/123/vdcs/vdc232/vm34/vol139", "name" : "VOL 2" }
1083         ]
1084     } ,
1085     "interfaces" : {
1086         "uri" : "/123/vdcs/vdc232/vm34/interfaces" ,
1087         "type" : "NetworkInterface" ,
1088         "total" : "2" ,
1089         "elements" : [
1090             { "uri" : "/123/vdcs/vdc232/vm34/int95", "name" : "Private
1091 Non-Routable" } ,
1092             { "uri" : "/123/vdcs/vdc232/vm34/int89", "name" :
1093 "Routable" }
1094         ]
1095     } ,
1096     "params" : {
1097         ...
1098     } ,
1099     "archives" : {
1100         "uri" : "/123/vdcs/vdc232/vm34/archives" ,
1101         "type" : "Archive" ,
1102         "total" : "2" ,
1103         "elements" : [
1104             { "uri" : "/123/vdcs/vdc232/vm34/sn1", "name" : "Archive -
1105 03012010" } ,
1106             { "uri" : "/123/vdcs/vdc232/vm34/sn2", "name" : "Archive -
1107 03022010" }
1108         ]
1109     } ,
1110     "resource state" : {
1111         "state" : "READY"
1112     } ,
1113     "created" : "February 28, 2010 15:34:34 GMT" ,
1114     "type" : "VIRTUAL"
1115 }

```

1116

1117 13.2 Resize VM

1118 To resize a VM, the client can PUT the desired *cpu*, *memory*, to the VM that may be different than the
 1119 current sizes. A service provider may additionally support using *params:size*. If a VM cannot be resized,
 1120 status 406 (Not Acceptable) may be returned.

1121

1122 **Synopsis:** PUT {URI of the VM}

1123 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 1124 Authorization, Accept

1125 **Request Message Body:** VM

1126 **Response Headers:** Content-Length, Content-Type, Content-Location

1127 **Response Message Body:** Server

1128 **Response Status:** 200, 202, 400, 401, 403, 406, 404, 409

1129

1130 **Example Request:** resize VM by modifying cpu, memory, disk

```
1131 PUT /123/vdcs/vdc232/vm32
1132 Host: cloudcompany.com
1133 Authorization: Basic xxxxxxxxxxxx
1134 Content-Length: nnn
1135 Content-Type: application/vnd.com.oracle.cloud.VM+json
1136 Accept: application/vnd.com.oracle.cloud.VM+json?resourcest*
1137 X-Cloud-Client-Specification-Version: 0.1
1138 {
1139   "cpu" : "3200" ,
1140   "memory" : "4096"
1141 }
```

1142

1143 **Example Response:**

```
1144 HTTP/1.1 202 Accepted
1145 Content-Type: application/vnd.com.oracle.cloud.VM+json
1146 Content-Location: /123/vdcs/vdc232/vm34
1147 Content-Length: nnn
1148 {
1149   "uri" : "/123/vdcs/vdc232/vm34",
1150   "resource state" : {
1151     "state" : "RESIZING",
1152     "progress" : "40",
1153     "message" : [
1154       { "code" : "1", "text" : "shut down VM" },
1155       { "code" : "5", "text" : "take snapshot" },
1156       ...
1157     ]
1158   }
1159 }
```

1160

1161 **Example Request:** (optionally) resize Server by PUTting to params

```
1162 PUT /123/vdcs/vdc232/vm34
1163 Host: cloudcompany.com
1164 Authorization: Basic xxxxxxxxxxxx
1165 Content-Length: nnn
1166 Content-Type: application/vnd.com.oracle.cloud.VM+json
1167 Accept: application/vnd.com.oracle.cloud.VM+json?resourcest*
1168 X-Cloud-Client-Specification-Version: 0.1
1169 {
1170   "params" : {
1171     "size" : "LARGE-PERFORMANCE"
1172   }
1173 }
```

1174

1175 **Example Response:**

```
1176 HTTP/1.1 202 Accepted
```

```

1177 Content-Type: application/vnd.com.oracle.cloud.VM+json
1178 Content-Location: /123/vdcs/vdc232/vm34
1179 Content-Length: nnn
1180 {
1181   "uri" : "/123/vdcs/vdc232/vm34",
1182   "resource state" : {
1183     "state" : "RESIZING",
1184     "progress" : "40",
1185     "message" : [
1186       { "code" : "1", "text" : "shut down VM" },
1187       { "code" : "5", "text" : "take snapshot" },
1188       ...
1189     ]
1190   }
1191 }

```

1192

1193 13.3 Control VM status

1194 A VM's status can be changed with a PUT to start, stop, suspend, resume, and restart the operational
 1195 status. There are currently 3 steady state allowed for a VM: (STOPPED, STARTED, SUSPENDED). The
 1196 following outlines what a service provider may observe:

- 1197 ▪ From STOPPED to STARTED : post STARTING, STARTED, RESUMING,
- 1198 RESTARTING
- 1199 ▪ From STARTED to STOPPED : post STOPPING, STOPPED
- 1200 ▪ From STARTED to STARTED : post RESTARTING (to mean stop and then start)
- 1201 ▪ From STARTED to SUSPENDED: post SUSPENDED, SUSPENDING
- 1202 ▪ From SUSPENDED to STARTED: post STARTED, RESUMING, STARTING
- 1203 ▪ From SUSPENDED to STOPPED: post STOPPED, STOPPING

1204 All other transitions may be ignored by the platform with 200, or returned with 409. For example, a VM may
 1205 prohibit the transition from STOPPED to SUSPENDED.

1206

1207 If there are topology specifications that depend on the VM to be up and running, the server may return 421
 1208 (Dependency Not Allowed) when the request is trying to shutdown the VM.

1209

1210 **Synopsis:** PUT {URI of the VM}

1211 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 1212 Authorization, Accept

1213 **Request Message Body:** VM

1214 **Response Headers:** Content-Length, Content-Type, Content-Location

1215 **Response Message Body:** VM

1216 **Response Status:** 200, 202, 400, 401, 403, 404, 409, 421

1217

1218 **Example Request:** control the Server operational status

```

1219 PUT /123/vdcs/vdc232/vm34
1220 Host: cloudcompany.com
1221 Authorization: Basic xxxxxxxxxxxx
1222 Content-Length: nnn

```



```

1223 Content-Type: application/vnd.com.oracle.cloud.VM+json
1224 Accept: application/vnd.com.oracle.cloud.VM+json?resourcest*
1225 X-Cloud-Client-Specification-Version: 0.1
1226 {
1227   "status" : "STOPPED"
1228 }

```

1229

1230 Example Response:

```

1231 HTTP/1.1 202 Accepted
1232 Content-Type: application/vnd.com.oracle.cloud.Server+json
1233 Content-Location: /123/vdcs/vdc232/vm34
1234 Content-Length: nnn
1235 {
1236   "uri" : "/123/vdcs/vdc232/vm34",
1237   "resource state" : {
1238     "state" : "STOPPING",
1239     "progress" : "60",
1240     "message" : [
1241       { "code" : "1", "text" : "shutting down VM" } ,
1242       ...
1243     ]
1244   }
1245 }

```

1246

1247 13.4 Add or Remove Network interfaces

1248 Network interfaces can be added to a VM via a POST to the Collection resource. The platform may
 1249 enforce the network integrity by making sure that each network interface cannot be associated with multiple
 1250 Server's. However, it is not required by the API.

1251

1252 **Synopsis:** POST {URI of the VM}

1253 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 1254 Authorization, Accept

1255 **Request Message Body:** NetworkInterface1256 **Response Headers:** Content-Length, Content-Type, Content-Location1257 **Response Message Body:** VM1258 **Response Status:** 200, 202, 400, 401, 403, 404, 406, 409, 421

1259

1260 Example Request: adding a Network instance to a Server, presuming the Network instance is already
 1261 created

```

1262 POST /123/vdcs/vdc232/vm34
1263 Host: cloudcompany.com
1264 Authorization: Basic xxxxxxxxxxxx
1265 Content-Length: nnn
1266 Content-Type: application/vnd.com.oracle.cloud.VNet+json
1267 Accept: application/vnd.com.oracle.cloud.VM+json?resourcest*
1268 X-Cloud-Client-Specification-Version: 0.1
1269 {
1270   "uri" : "/123/vdcs/vdc232/net91/int3"

```

1271

}

1272

1273 Example Response:

```

1274 HTTP/1.1 202 Accepted
1275 Content-Type: application/vnd.com.oracle.cloud.Server+json
1276 Content-Location: /123/vdcs/vdc232/vm34
1277 Content-Length: nnn
1278 {
1279   "uri" : "/123/vdcs/vdc232/vm34",
1280   "resource state" : {
1281     "state" : "MODIFYING NETWORK",
1282     "progress" : "80",
1283     "message" : [
1284       { "code" : "1", "text" : "attaching network" } ,
1285       ...
1286     ]
1287   }
1288 }

```

1289

1290 After the resourcestatus.state is in READY state, the /123/vdcs/vdc232/vm34?networks may look like the
 1291 following:

1292

1293 Example Response: after the network interface has been added

```

1294 HTTP/1.1 202 Accepted
1295 Content-Type: application/vnd.com.oracle.cloud.Server+json
1296 Content-Location: /123/vdcs/vdc232/vm34
1297 Content-Length: nnn
1298 {
1299   "uri" : "/123/vdcs/vdc232/vm34",
1300   "interfaces" : {
1301     "uri" : "/123/vdcs/vdc232/vm34/interfaces",
1302     "type" : "NetworkInterface",
1303     "total" : "3" ,
1304     "elements" : [
1305       { "uri" : "/123/vdcs/vdc232/vm34/int95", "name" : "Private
1306 Non-Routable" },
1307       { "uri" : "/123/vdcs/vdc232/vm34/int89", "name" :
1308 "Routable" } ,
1309       { "uri" : "/123/vdcs/vdc232/vm34/int98", "name" : "Routable
1310 From EM Testing Network Profile" }
1311     ]
1312   }
1313 }
1314 }

```

1315

1316 It is important to note that the URI of the added network "/123/vdcs/vdc232/vm34/int98" is different from the
 1317 "/123/vdcs/vdc232/net91/int3". In other words, the URI in the VM scope is a reference to the actual
 1318 resource. Thus, deleting a network interface from a VM is logically equivalent to removing the association,
 1319 and not deleting the actual network interface resource.

1320

1321 **Synopsis:** DELETE {URI of the Network Interface in a VM}

1322 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
1323 Authorization, Accept

1324 **Request Message Body:** N/A

1325 **Response Headers:** Content-Length, Content-Type, Content-Location

1326 **Response Message Body:** VM

1327 **Response Status:** 200, 202, 400, 401, 403, 404, 406, 409, 421

1328

1329 Example Request: deleting a network interface from a Server

```
1330 DELETE /123/vdcs/vdc232/vm34/int98
1331 Host: cloudcompany.com
1332 Authorization: Basic xxxxxxxxxxxx
1333 Content-Length: nnn
1334 Content-Type: application/vnd.com.oracle.cloud.VNet+json
1335 Accept: application/vnd.com.oracle.cloud.VM+json?resourcest*
1336 X-Cloud-Client-Specification-Version: 0.1
```

1337

1338 Example Response:

```
1339 HTTP/1.1 202 Accepted
1340 Content-Type: application/vnd.com.oracle.cloud.VM+json
1341 Content-Location: /123/vdcs/vdc232/vm34
1342 Content-Length: nnn
1343 {
1344   "uri" : "/123/vdcs/vdc232/vm34",
1345   "resource state" : {
1346     "state" : "MODIFYING NETWORK",
1347     "progress" : "80",
1348     "message" : [
1349       {"code" : "1", "text" : "detaching network"} ,
1350       ...
1351     ]
1352   }
1353 }
```

1354

1355 13.5 Adding or Removing Volume

1356 Adding a volume to a VM is very similar to adding a network to a Server, and thus, not repeated here.

1357

1358 However, the service provider may consider the following while deleting a volume from a VM

- 1359 ▪ If the volume is shared among multiple entities, then, removing the "association" only,
1360 and not the actual volume
- 1361 ▪ If the volume is dedicate to this VM, then, deleting the volume may remove the actual
1362 underlying volume and its data

1363 13.6 Creating a VM Archive

1364 The service provider may consider the following when accepting a VM archive request

- 1365 ▪ The scope of the archive could be shallow, meaning that not all volumes would be
1366 archived, and this is the default behavior

- 1367 ▪ The scope of the snapshot could be deep, meaning that all volumes attached to the VM
1368 would also be snapshot

1369 The service provider can use the *params* to indicate which mode, and should publish its default behavior if
1370 it is different from the above.

1371

1372 **Synopsis:** POST {URI of the VM}

1373 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
1374 Authorization, Accept

1375 **Request Message Body:** Archive

1376 **Response Headers:** Content-Length, Content-Type, Content-Location

1377 **Response Message Body:** Archive

1378 **Response Status:** 200, 202, 400, 401, 403, 404, 406, 409

1379

1380 Example Request: Archive a VM

```
1381 POST /123/vdcs/vdc232/vm34
1382 Host: cloudcompany.com
1383 Authorization: Basic xxxxxxxxxxxx
1384 Content-Length: nnn
1385 Content-Type: application/vnd.com.oracle.cloud.Archive+json
1386 Accept: application/vnd.com.oracle.cloud.Archive+json
1387 X-Cloud-Client-Specification-Version: 0.1
1388 {
1389   "name" : "Archive-03-03-2010" ,
1390   "description" : "Taking an archiveafter some software
1391 configurations ..." ,
1392   "param" : {
1393     "mode" : "THIN"
1394   }
1395 }
```

1396 Example Response:

```
1397 HTTP/1.1 202 Accepted
1398 Content-Type: application/vnd.com.oracle.cloud.Archive+json
1399 Content-Location: /123/vdcs/vdc232/vm34/archives/sn4
1400 Content-Length: nnn
1401 {
1402   "uri" : "/123/vdcs/vdc232/vm34/archives/sn4" ,
1403   "name" : "ARchive-03-03-2010" ,
1404   "description" : "Taking a snapshot after some software
1405 configurations ..." ,
1406   "created" : "March 3, 2010 12:34:22 GMT" ,
1407   "source" : "/123/vdcs/vdc232/vm34",
1408   "sourcetype" : "VM" ,
1409   "resource state" : {
1410     "state" : "CREATING",
1411     "progress" : "30" ,
1412     "message" : [
1413       {"code" : "5", "text" : "copying bits of..."},
1414       ...
1415     ]
1416   }
1417   "params" : {
1418     "mode" : "THIN"
```

```
1419     }
1420 }
```

1421

1422

1423 [Return to Section List](#)

1424

1425 14 Operations on Volume resources

1426 Operations on a Volume resource allows clients to

- 1427 ▪ Inquire the detail attributes
- 1428 ▪ Create a new Volume via a storage template or via a snapshot
- 1429 ▪ Resize and modify a Volume
- 1430 ▪ Create a snapshot

1431 14.1 Get Volume

1432 Retrieving the details of a Volume.

1433

1434 **Synopsis:** GET {URI of a Volume}

1435 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

1436 **Request Message Body:** N/A.

1437 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

1438 **Response Message Body:** Volume

1439 **Response Status:** 200, 400, 401, 403, 404, 410

1440

1441 **Example Request:** Retrieve the Volume attributes

```
1442 GET /123/vdcs/vdc232/vol142
1443 Host: cloudcompany.com
1444 Authorization: Basic xxxxxxxxxxxx
1445 Accept: application/vnd.com.oracle.cloud.Volume+json
1446 X-Cloud-Client-Specification-Version: 0.1
```

1447

1448 **Example Response**

```
1449 HTTP/1.1 200 OK
1450 Content-Type: application/vnd.com.oracle.cloud.Volume+json
1451 Content-Location: /123/vdcs/vdc232/vol142
1452 Cache-Control: no-store
1453 Content-Length: nnn
1454 {
1455   "uri" : "...",
1456   "name" : "Storage for Software Library",
1457   "description" : "A shared storage storing software library
1458 entities to be shared between the OMS's..." ,
1459   "tags" : ["Testing", "SWLib", "Shared", "OMS", ... ] ,
```

```

1460     "contained in" : "/123/vdcs/vdc232" ,
1461     "container type" : "VDC" ,
1462     "based on" : "/templtaes/items/t23" ,
1463     "params" : {
1464         "can snapshot" : "TRUE" ,
1465         "can thin clone" : "TRUE" ,
1466         "raid type" : "RAID 5" ,
1467         ...
1468     } ,
1469     "size" : "512" ,
1470     "snapshots" : {
1471         "uri" : "/123/vdcs/vdc232/vol142/snapshots" ,
1472         "type" : "Snapshot" ,
1473         "total" : "2" ,
1474         "elements" : [
1475             { "uri" : "/123/vdcs/vdc232/vol142/sn1" , "name" : "Snapshot
1476 - 03012010" } ,
1477             { "uri" : "/123/vdcs/vdc232/vol142/sn2" , "name" : "Snapshot
1478 - 03022010" }
1479         ]
1480     } ,
1481     "resource state" : {
1482         "state" : "READY"
1483     } ,
1484     "created" : "February 28, 2010 15:34:34 GMT"
1485 }

```

1486

1487 14.2 Create a new Volume

1488 A volume can be created into an existing VDC.

1489 The service provider may permits "short-cuts" that allows the client to

- 1490 ▪ Directly create a volume to a Server

1491

1492 **Synopsis:** POST {URI of the VDC}

1493 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
1494 Authorization, Accept

1495 **Request Message Body:** Volume

1496 **Response Headers:** Content-Length, Content-Type, Content-Location

1497 **Response Message Body:** Volume

1498 **Response Status:** 200, 202, 400, 401, 403, 404, 409

1499

1500 **Example Request:** Create a new volume based on a storage template

```

1501 POST /123/vdcs/vdc232
1502 Host: cloudcompany.com
1503 Authorization: Basic xxxxxxxxxxxx
1504 Content-Length: nnn
1505 Content-Type: application/vnd.com.oracle.cloud.Volume+json
1506 Accept: application/vnd.com.oracle.cloud.Volume+json
1507 X-Cloud-Client-Specification-Version: 0.1
1508 {

```

```

1509     "name" : "Additional Software Library Storage",
1510     "description" : "This is an additional storage for software
1511 library uses..." ,
1512     "tags" : ["Storage", "RAID 5", "Software Library" ...] ,
1513     "based on" : "/templates/items/t23" ,
1514     "params" : {
1515         "can snapshot" : "TRUE" ,
1516         "can clone" : "TRUE" ,
1517         "raid type" : "RAID 5",
1518         ...
1519     }
1520     "size" : "1024"
1521 }

```

1522

Example Response:

```

1524 HTTP/1.1 202 Accepted
1525 Content-Type: application/vnd.com.oracle.cloud.Volume+json
1526 Content-Location: /123/vdcs/vdc232/vol149
1527 Content-Length: nnn
1528 {
1529     "uri" : "/123/vdcs/vdc232/vol149",
1530     "resource state" : {
1531         "state" : "CREATING",
1532         "progress" : "40",
1533         "message" : [
1534             { "code" : "1", "text" : "creating" },
1535             ...
1536         ]
1537     }
1538 }

```

1539

Example Request: Create a new volume based on an existing volume snapshot

```

1541 POST /123/vdcs/vdc232
1542 Host: cloudcompany.com
1543 Authorization: Basic xxxxxxxxxxxx
1544 Content-Length: nnn
1545 Content-Type: application/vnd.com.oracle.cloud.Volume+json
1546 Accept: application/vnd.com.oracle.cloud.Volume+json
1547 X-Cloud-Client-Specification-Version: 0.1
1548 {
1549     "name" : "Additional Software Library Storage",
1550     "description" : "This is an additional storage for software
1551 library uses..." ,
1552     "params" : {
1553         "can snapshot" : "TRUE" ,
1554         "can thin clone" : "TRUE" ,
1555         "raid type" : "RAID 5",
1556         "clone type" : "THIN" ,
1557         ...
1558     }
1559     "clone from" : "/123/vdcs/vdc232/vol142/sn2" ,
1560     "size" : "1024"
1561 }

```

1562

Example Response:

```

1564 HTTP/1.1 202 Accepted
1565 Content-Type: application/vnd.com.oracle.cloud.Volume+json

```

```

1566 Content-Location: /123/vdcs/vdc232/vol152
1567 Content-Length: nnn
1568 {
1569   "uri" : "/123/vdcs/vdc232/vol152",
1570   "resource state" : {
1571     "state" : "CREATING",
1572     "progress" : "40",
1573     "message" : [
1574       { "code" : "1", "text" : "creating thin clone" },
1575       ...
1576     ]
1577   }
1578 }

```

1579

1580 [Return to Section List](#)

1581

1582 14.3 Resize Volume

1583 To resize a Volume, the client can do a PUT with the desired *size* for the Volume that may be different than
 1584 the current size. A service provider may consider

- 1585 ▪ Permitting going from a fixed size to a unlimited size
- 1586 ▪ Prohibits a volume with a unlimited size to become a fixed size

1587

1588 **Synopsis:** PUT {URI of the Volume}

1589 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 1590 Authorization, Accept

1591 **Request Message Body:** Volume1592 **Response Headers:** Content-Length, Content-Type, Content-Location1593 **Response Message Body:** VM1594 **Response Status:** 200, 202, 400, 401, 403, 404, 409

1595

1596 **Example Request:** resize Volume

```

1597 PUT /123/vdcs/vdc232/vol142
1598 Host: cloudcompany.com
1599 Authorization: Basic xxxxxxxxxxxx
1600 Content-Length: nnn
1601 Content-Type: application/vnd.com.oracle.cloud.Volume+json
1602 Accept: application/vnd.com.oracle.cloud.Volume+json
1603 X-Cloud-Client-Specification-Version: 0.1
1604 {
1605   "size" : "1024"
1606 }

```

1607

1608 **Example Response:**

```

1609 HTTP/1.1 200 Ok
1610 Content-Type: application/vnd.com.oracle.cloud.Volume+json
1611 Content-Location: /123/vdcs/vdc232/vol142

```



```

1612 Content-Length: nnn
1613 {
1614   "uri" : "/123/dg/34/vol142",
1615   "resource state" : {
1616     "state" : "READY",
1617     "progress" : "100",
1618     "message" : [
1619       { "code" : "1", "text" : "resized" },
1620       ...
1621     ]
1622   }
1623 }

```

1624

1625 14.4 Creating a Volume Snapshot

1626 Creating a volume snapshot is very similar to other type of snapshots

1627

1628 **Synopsis:** POST {URI of the Volume}

1629 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 1630 Authorization, Accept

1631 **Request Message Body:** Snapshot

1632 **Response Headers:** Content-Length, Content-Type, Content-Location

1633 **Response Message Body:** Snapshot

1634 **Response Status:** 200, 202, 400, 401, 403, 404, 409

1635

1636 Example Request: Snapshot a Volume

```

1637 POST /123/vdcs/vdc232/vol142
1638 Host: cloudcompany.com
1639 Authorization: Basic xxxxxxxxxxxx
1640 Content-Length: nnn
1641 Content-Type: application/vnd.com.oracle.cloud.Snapshot+json
1642 Accept: application/vnd.com.oracle.cloud.Snapshot+json
1643 X-Cloud-Client-Specification-Version: 0.1
1644 {
1645   "name" : "Snapshot-03-03-2010" ,
1646   "description" : "Taking a snapshot after some software library
1647   entries added ..." ,
1648   "source" : "/123/vdcs/vdc232/vol142" ,
1649   "sourcetype" : "Volume" ,
1650   "param" : {
1651     ...
1652   }
1653 }

```

1654

1655 Example Response:

```

1656 HTTP/1.1 202 Accepted
1657 Content-Type: application/vnd.com.oracle.cloud.Snapshot+json
1658 Content-Location: /123/vdcs/vdc232/vol142/sn3
1659 Content-Length: nnn

```

```

1660 {
1661   "uri" : "/123/vdcs/vdc232/vol142/sn3" ,
1662   "name" : "Snapshot-03-03-2010" ,
1663   "description" : "Taking a snapshot after some software library
1664   entries added ..." ,
1665   "created" : "March 3, 2010 14:24:22 GMT" ,
1666   "source" : "/123/vdcs/vdc232/vol142" ,
1667   "sourcetype" : "Volume" ,
1668   "resource state" : {
1669     "state" : "CREATING" ,
1670     "progress" : "30" ,
1671     "message" : [
1672       { "code" : "5", "text" : "taking snapshot of..." },
1673     ]
1674   }
1675 }
1676 }

```

1677

1678

1679 [Return to Section List](#)

1680

1681 15 Operations on Archive resources

1682 Archive creations have been described in VM resource above, and thus would not be duplicated here. The
 1683 service provider may consider the following when managing the lifecycle of an archive

- 1684 ▪ After an archive is deleted, if there were any VM's that were created based on a archive,
 1685 dereferencing the archive URI should return 410

1686

1687 15.1 Get Archive

1688 Retrieving the details of an Archive.

1689

1690 **Synopsis:** GET {URI of a Archive}

1691 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

1692 **Request Message Body:** N/A.

1693 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

1694 **Response Message Body:** Archive

1695 **Response Status:** 200, 400, 401, 403, 404, 410

1696

1697 Example Request: Retrieve the Archive attributes

```

1698 GET /123/vdcs/vdc232/vm34/sn2
1699 Host: cloudcompany.com
1700 Authorization: Basic xxxxxxxxxxxx
1701 Accept: application/vnd.com.oracle.cloud.Archive+json

```

1702 X-Cloud-Client-Specification-Version: 0.1

1703

1704 Example Response

```

1705 HTTP/1.1 200 OK
1706 Content-Type: application/vnd.com.oracle.cloud.Archive+json
1707 Content-Location: /123/vdcs/vdc232vm34/sn2
1708 Cache-Control: no-store
1709 Content-Length: nnn
1710 {
1711   "uri" : "...",
1712   "name" : "Archive - 03022010",
1713   "description" : "Archive taken after reconfiguring the Web
1714 server..." ,
1715   "created" : "March 2, 2010 10:14:36 GMT" ,
1716   "source" : "/123/vdcs/vdc232/vm34" ,
1717   "sourcetype" : "VM" ,
1718   "params" : {
1719     "mode" : "THIN"
1720   } ,
1721   "resource state" : {
1722     "state" : "READY"
1723   }
1724 }

```

1725

1726 15.2 Rolling back a resource to an archive

1727 Rolling back a resource means replacing the resource with an archive that was taken. There are some
 1728 versioning complexities. However, the service provider MAY consider the following:

- 1729 ■ When rolling back a resource back to an archive, the list of archives is not rolled back.
 1730 For example, if VM has archives (s1,s2,s3) in chronological order, then rolling back VM
 1731 to s2 means that the replaced resource would still have (s1, s2, s3) in its snapshot
 1732 lists
- 1733 ■ It is permissible for a resource to roll back to the same snapshot again and again
- 1734 ■ When additional snapshots are taken, the service provider SHALL uniquely name each
 1735 of the new branches

1736

1737 The following example would rollback vm34 to sn2

1738

1739 **Synopsis:** PUT {URI of the VM}

1740 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 1741 Authorization, Accept

1742 **Request Message Body:** VM

1743 **Response Headers:** Content-Length, Content-Type, Content-Location

1744 **Response Message Body:** Server

1745 **Response Status:** 200, 202, 400, 401, 403, 404, 406, 409

1746

1747 Example Request: rollback a VM

```

1748 PUT /123/vdcs/vdc232/vm34
1749 Host: cloudcompany.com
1750 Authorization: Basic xxxxxxxxxxxx
1751 Content-Length: nnn
1752 Content-Type: application/vnd.com.oracle.cloud.VM+json
1753 Accept: application/vnd.com.oracle.cloud.VM+json
1754 X-Cloud-Client-Specification-Version: 0.1
1755 {
1756   restored from: {
1757     "uri" : "/123/vdcs/vdc232/vm34/sn2"
1758   }
1759 }

```

1760

1761 Example Response:

```

1762 HTTP/1.1 202 Accepted
1763 Content-Type: application/vnd.com.oracle.cloud.VM+json
1764 Content-Location: /123/vdcs/vdc232/vm34
1765 Cache-Control: no-store
1766 Content-Length: nnn
1767 {
1768   "uri" : "...",
1769   "name" : "Simple Web Server for serving Reports",
1770   "description" : "A single Web Server that can serve the QA
1771 testing reports and ...",
1772   "tags" : ["Testing", "Web Server", "EM", "Reporting", ... ] ,
1773   "contained in" : "/123/vdcs/vdc232" ,
1774   "container type" : "VDC" ,
1775   "based on" : "/templates/items/t904" ,
1776   "hostname" : "qareport234.mycompany.com" ,
1777   "cpu" : "2400" ,
1778   "memory" : "2048" ,
1779   "disks" : [
1780     ["/mnt/disk1", "128"]
1781   ] ,
1782   ...
1783   "archives" : {
1784     "uri" : "/123/vdcs/vdc232/vm34/archives",
1785     "type" : "Archives",
1786     "total" : "3" ,
1787     "elements" : [
1788       { "uri" : "/123/vdcs/vdc232/vm34/sn1", "name" : "Archive -
1789 03012010" },
1790       { "uri" : "/123/vdcs/vdc232/vm34/sn2", "name" : "Archive -
1791 03022010" },
1792       { "uri" : "/123/vdcs/vdc232/vm34/sn3", "name" : "Archive -
1793 03032010" }
1794     ]
1795   } ,
1796   "restored from" : {
1797     "uri" : "/123/vdcs/vdc232/vm34/sn2" ,
1798     "name" : "Archive - 03022010"
1799   }
1800   "resource state" : {
1801     "state" : "BEING ROLLBACK",
1802     "progress" : "50" ,
1803     "message" : [
1804       { "code" : "2" , "text" : "rolling back VM..." }
1805     ]
1806   }
1807 }

```

1808

1809 It is important to note the following

- 1810 ▪ PUT request would change the VM in-place and no new resource is generated
- 1811 ▪ The archives of the rolled back VM still contains sn3

1812

1813

1814 [Return to Section List](#)

1815

1816 16 Operations on VNet resources

1817 Operations on a vnet resource permits the client to

- 1818 ▪ Create a vnet based on some network service template
- 1819 ▪ Modify a vnet's configuration

1820

1821 A service provider may consider virtual network to be hardcoded assets in the cloud. For example, a
 1822 service provider may provide "Load Balancer service" where a new instance of a "Load Balancer" can be
 1823 instantiated to be included in a VDC. In that case, the "Load Balancer service" is the network template,
 1824 while an instance of the "Load Balancer" is a VNet.

1825

1826 Similarly, a network may be created as a vnet that provides IP addresses, for example, a DHCP server.

1827

1828 16.1 Get VNet

1829 Retrieving the details of a VNet.

1830

1831 **Synopsis:** GET {URI of a VNet}1832 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization1833 **Request Message Body:** N/A.1834 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location1835 **Response Message Body:** VNet1836 **Response Status:** 200, 400, 401, 403, 404, 410

1837

1838 Example Request: Retrieve the VNet attributes

```

1839 GET /123/vdcs/vdc232/net91
1840 Host: cloudcompany.com
1841 Authorization: Basic xxxxxxxxxxxx
1842 Accept: application/vnd.com.oracle.cloud.VNet+json
1843 X-Cloud-Client-Specification-Version: 0.1

```

1844

1845 Example Response

```

1846 HTTP/1.1 200 OK
1847 Content-Type: application/vnd.com.oracle.cloud.VNet+json
1848 Content-Location: /123/vdcs/vdc232//net91
1849 Cache-Control: no-store
1850 Content-Length: nmn
1851 {
1852   "uri" : "...",
1853   "name" : "Routable Network Profile for Testing EM",
1854   "description" : "A Network profile that would contain routable
1855 network instances assigned to other resource..." ,
1856   "tags" : ["Testing", "Routable", "VNet", ... ] ,
1857   "contained in" : "/123/vdcs/vdc232" ,
1858   "container type" : "VDC" ,
1859   "based on" : "/templates/items/n4" ,
1860   "params" : {
1861     "ip range" : "192.168.1.0 - 192.168.1.199" ,
1862     "subnet" : "255.255.255.0" ,
1863     ...
1864   } ,
1865   "created" : "February 28, 2010 15:34:34 GMT" ,
1866   "interfaces" : {
1867     "uri" : "/123/vdcs/vdc232/net91/interfaces",
1868     "type" : "NetworkInterface",
1869     "total" : "2" ,
1870     "elements" : [
1871       { "uri" : "/123/vdcs/vdc232/net91/int1", "name" :
1872 "192.168.0.1" },
1873       { "uri" : "/123/vdcs/vdc232/net91/int2", "name" :
1874 "192.168.0.2" }
1875     ]
1876   }
1877 }

```

1878

1879 **16.2 Create a VNet**

1880 A network can be created based on a network template. A service provider may implicitly create network
 1881 resource during creation of VDC.

1882

1883 **Synopsis:** POST {URI of the VDC}

1884 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 1885 Authorization, Accept

1886 **Request Message Body:** VNet1887 **Response Headers:** Content-Length, Content-Type, Content-Location1888 **Response Message Body:** VNet1889 **Response Status:** 200, 202, 400, 401, 403, 404, 409

1890

1891 Example Request: Create a new network based on a network template

1892 `POST /123/vdcs/vdc232`

```

1893 Host: cloudcompany.com
1894 Authorization: Basic xxxxxxxxxxxx
1895 Content-Length: nnn
1896 Content-Type: application/vnd.com.oracle.cloud.VNet+json
1897 Accept: application/vnd.com.oracle.cloud.VNet+json?resource state
1898 X-Cloud-Client-Specification-Version: 0.1
1899 {
1900   "name" : "Load Balancer - Web Servers",
1901   "description" : "This is a load balancer for the web services..." ,
1902   "tags" : ["Load Balancer", "Web servers" ...] ,
1903   "based on" : "/templates/items/ne6" ,
1904   "params" : {
1905     ...
1906   }
1907 }

```

1908

1909 **Example Response:**

```

1910 HTTP/1.1 202 Accepted
1911 Content-Type: application/vnd.com.oracle.cloud.VNet+json
1912 Content-Location: /123/vdcs/vdc232/net98
1913 Content-Length: nnn
1914 {
1915   "uri" : "/123/vdcs/vdc232/net98",
1916   "resource state" : {
1917     "state" : "CREATING",
1918     "progress" : "40",
1919     "message" : [
1920       { "code" : "1", "text" : "creating" },
1921       ...
1922     ]
1923   }
1924 }

```

1925

1926

1927 16.3 Delete a VNet

1928 Deleting a network would cascade the deletion of the network instances that are contained. Thus, if there
 1929 were resources associated with a network instance, the server may return 421 status.

1930

1931 Similarly, if a network is part of a topology that is connected to other resources, the provider may enforce
 1932 the dependency by prohibiting the deletion (or modification) of the network resource.

1933

1934 **Synopsis:** DELETE {URI of a VNet}

1935 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

1936 **Request Message Body:** N/A

1937 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

1938 **Response Message Body:** N/A

1939 **Response Status:** 200, 400, 401, 403, 404, 410

Oracle Cloud Management API

1940

1941 Example Request: deleting a network

```
1942 DELETE /123/vdcs/vdc232/net91
1943 Host: cloudcompany.com
1944 Authorization: Basic xxxxxxxxxxxx
1945 Accept: application/vnd.com.oracle.cloud.VNet+json
1946 X-Cloud-Client-Specification-Version: 0.1
```

1947

1948 Example Response

```
1949 HTTP/1.1 200 OK
```

1950

1951

1952 [Return to Section List](#)

1953

1954

1955 **17 Operations on NetworkInterface resources**

1956 Operations on a network interface resource include

- 1957 ▪ Inquire the attributes of a network interface
- 1958 ▪ Create a network interface
- 1959 ▪ Delete a network interface

1960 A service provider may restrict how a network interface can be modified once created. It is not common for
 1961 a network interface to be changed.

1962

1963 **17.1 Get Network Interface**

1964 Retrieving the details of a Network Interface.

1965

1966 **Synopsis:** GET {URI of a NetworkInterface}1967 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization1968 **Request Message Body:** N/A.1969 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location1970 **Response Message Body:** NetworkInterface1971 **Response Status:** 200, 400, 401, 403, 404, 410

1972

1973 Example Request: Retrieve the Network interface attributes

```

1974 GET /123/vdcs/vdc232/net91/int1
1975 Host: cloudcompany.com
1976 Authorization: Basic xxxxxxxxxxxx
1977 Accept: application/vnd.com.oracle.cloud.NetworkInterface+json
1978 X-Cloud-Client-Specification-Version: 0.1

```

1979

1980 Example Response

```

1981 HTTP/1.1 200 OK
1982 Content-Type:
1983 application/vnd.com.oracle.cloud.NetworkInterface+json
1984 Content-Location: /123/vdcs/vdc232/net91/int1
1985 Cache-Control: no-store
1986 Content-Length: nnn
1987 {
1988   "uri" : "...",
1989   "name" : "192.168.0.1",
1990   "routable" : "TRUE",
1991   "vnet" : "/123/vdcs/vdc232/net91" ,
1992   "address" : "192.168.0.1"
1993 }

```

1994

1995 **17.2 Create Network Interface**

1996 To create a network interface, the client can post to a VNet that is capable of generating a network
 1997 interface. For example, using a virtual network that contains IP addresses that can be allocated and
 1998 assigned. If the client's attempt to post to a virtual network that is not capable of generating a network
 1999 instance, 406 (not acceptable) may be returned.

2000

2001 It is permissible to post an empty message body to generate a network interface. The server SHALL
 2002 ensure the network interface does not conflict and cause duplicated network identities. Error 406 (not
 2003 acceptable) is returned otherwise.

2004

2005 **Synopsis:** POST {URI of a VNet}

2006 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

2007 **Request Message Body:** NetworkInterface

2008 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

2009 **Response Message Body:** NetworkInterface

2010 **Response Status:** 200, 400, 401, 403, 404, 406, 410

2011

2012 **Example Request: Create a network interface**

```

2013 POST /123/vdcs/vdc232/net91
2014 Host: cloudcompany.com
2015 Authorization: Basic xxxxxxxxxxxx
2016 Content-Type:
2017 application/vnd.com.oracle.cloud.NetworkInterface+json
2018 Content-Length: nnn
2019 Accept: application/vnd.com.oracle.cloud.NetworkInterface+json
2020 X-Cloud-Client-Specification-Version: 0.1
2021 {
2022   "name" : "My Network Instance" ,
2023   "description" : "This network instance is going to be used for ..."
2024 }

```

2025

2026 **Example Response**

```

2027 HTTP/1.1 200 OK
2028 Content-Type:
2029 application/vnd.com.oracle.cloud.NetworkInterface+json
2030 Content-Location: /123/vdcs/vdc232/net91/int5
2031 Cache-Control: no-store
2032 Content-Length: nnn
2033 {
2034   "uri" : "...",
2035   "name" : "My Network Instance",
2036   "description" : "This network instance is going to be used for ..."
2037 ,
2038   "routable" : "TRUE",
2039   "vnet" : "/123/vdcs/vdc232/net91" ,
2040   "address" : "192.168.0.5"
2041 }

```

2042

2043 [Return to Section List](#)

2044

2045 **18 Operations on ScalabilityGroup resources**

2046 Requests to a scalability group allows the client to

- 2047 ▪ Retrieve attributes of the scalability groups
- 2048 ▪ Create a scalability group in a VDC
- 2049 ▪ Scale out or Scale in the scalability group
- 2050 ▪ Uniformly control the scalability group's status

2051

2052 **18.1 Get ScalabilityGroup**

2053 Retrieving the details of a Scalability group.

2054

2055 **Synopsis:** GET {URI of a ScalabilityGroup}2056 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization2057 **Request Message Body:** N/A.2058 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location2059 **Response Message Body:** ScalabilityGroup2060 **Response Status:** 200, 400, 401, 403, 404, 410

2061

2062 Example Request: Retrieve the ScalabilityGroup attributes

```

2063 GET /123/dg/34/c43
2064 Host: cloudcompany.com
2065 Authorization: Basic xxxxxxxxxxxx
2066 Accept: application/vnd.com.oracle.cloud.ScalabilityGroup+json
2067 X-Cloud-Client-Specification-Version: 0.1

```

2068

2069 Example Response

```

2070 HTTP/1.1 200 OK
2071 Content-Type:
2072 application/vnd.com.oracle.cloud.ScalabilityGroup+json
2073 Content-Location: /123/dg/34/c43
2074 Cache-Control: no-store
2075 Content-Length: nnn
2076 {
2077   "uri" : "...",
2078   "name" : "FMW Application Server - OMS",
2079   "description" : "The group of OMS nodes in the Enterprise
2080 Manager..." ,
2081   "nodes" : {
2082     "uri" : "/123/dg/34/c43/nodes",
2083     "type" : "URI",

```

```

2084     "total" : "4" ,
2085     "elements" : [
2086         { "uri" : "/123/dg/34/server633", "name" : "OMS 1" },
2087         { "uri" : "/123/dg/34/server634", "name" : "OMS 2" },
2088         { "uri" : "/123/dg/34/server635", "name" : "OMS 3" },
2089         { "uri" : "/123/dg/34/server636", "name" : "OMS 4" }
2090     ]
2091     },
2092     "type" : "Server" ,
2093     "count" : "4" ,
2094     "contained in" : "/123/dg/34" ,
2095     "container type" : "AssemblyInstance" ,
2096     "status" : "ONLINE" ,
2097     "tags" : [ "Testing", "OMS", "EM", "App Server", ... ] ,
2098     "resource state" : {
2099         "state" : "READY"
2100     } ,
2101     "created" : "February 26, 2010 15:34:34 GMT" ,
2102     "homogenous" : "TRUE" ,
2103     "based on" : "/templates/item/t232"
2104 }

```

2105

2106 18.2 Creates a new ScalabilityGroup

2107 A user can instantiate a new scalability group by posting to a VDC of which the scalability group is to
 2108 reside. The service provider SHALL permit a scalability group with 0 nodes be created and then the client
 2109 can populate it with other created resources.

2110

2111 A service provider may consider the following on homogenous scalability groups

- 2112 ▪ If the *min* is specified, when the scalability group is posted with *based_on* specified, the
- 2113 service provider may automatically initialize the scalability group to the *min* number of
- 2114 nodes

2115

2116 In this example, the user wants to add a new scalability group that will contain coherence nodes to the
 2117 VDC, "My Work Center".

2118

2119 **Synopsis:** POST {URI of the VDC}

2120 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 2121 Authorization, Accept

2122 **Request Message Body:** ScalabilityGroup

2123 **Response Headers:** Content-Length, Content-Type, Content-Location

2124 **Response Message Body:** ScalabilityGroup

2125 **Response Status:** 200, 202, 400, 401, 403, 404, 409

2126

2127 Example Request: creates a new scalability group

2128 `POST /123/vdcs/vdc232`

```

2129 Host: cloudcompany.com
2130 Authorization: Basic xxxxxxxxxxxx
2131 Content-Length: nnn
2132 Content-Type:
2133 application/vnd.oracle.cloud.ScalabilityGroup+json
2134 Accept: application/vnd.oracle.cloud.ScalabilityGroup+json
2135 X-Cloud-Client-Specification-Version: 0.1
2136 {
2137   "name" : "Coherence Node Cluster" ,
2138   "description" : "This cluster contains the coherence nodes where
2139 there will be an EM agent monitoring for testing..." ,
2140   "type" : "Server" ,
2141   "tags" : ["Coherence", "Testing", "EM", "Cluster"],
2142   "homogenous" : "TRUE" ,
2143   "based on" : "/templates/item/t234/",
2144   "min" : "2"
2145 }

```

2146

2147 Example Response:

```

2148 HTTP/1.1 200 Ok
2149 Content-Type:
2150 application/vnd.oracle.cloud.ScalabilityGroup+json
2151 Content-Location: /123/dg/34/c54
2152 Content-Length: nnn
2153 {
2154   "uri" : "/123/dg/34/c54",
2155   "name" : "Coherence Node Cluster",
2156   "description" : "This cluster contains the coherence nodes where
2157 there will be an EM agent monitoring for testing..." ,
2158   "nodes" : {
2159     "uri" : "/123/vdcs/vdc232/c54/nodes" ,
2160     "type" : "URI",
2161     "total" : "2" ,
2162     "elements" : [
2163       { "uri" : "/123/vdcs/vdc232/server653", "name" : "Node 1" },
2164       { "uri" : "/123/vdcs/vdc232/server654", "name" : "Node 2" }
2165     ]
2166   } ,
2167   "type" : "Server" ,
2168   "count" : "2" ,
2169   "contained in" : "/123/vdcs/vdc232" ,
2170   "container type" : "VDC" ,
2171   "status" : "CREATED"
2172   "tags" : ["Coherence", "Testing", "EM", "Cluster"],
2173   "resource state" : {
2174     "state" : "READY"
2175   } ,
2176   "created" : "March 2 2010, 01:24:34 GMT" ,
2177   "max" : "16" ,
2178   "min" : "2",
2179   "homogenous" : "TRUE",
2180   "based on" : "/menus/ovmtemplates/t234/",
2181 }

```

2182

2183 18.3 Shutdown the ScalabilityGroup

2184 To shutdown a scalability group, the user can do a PUT with status=OFFLINE to the Scalability group
 2185 resource. The server SHALL follow the topologies specifying the boot dependencies in the Assembly
 2186 Instance containing the scalability group in order to ensure dependencies are observed

2187

2188 A service provider may consider the following:

- 2189 ▪ If a scalability group is depended on by another entity, shutting down the Assembly
- 2190 Instance may result in 421 (Dependency Not Allowed) error
- 2191 ▪ If a scalability group is starting up when a shutdown request is submitted, the service
- 2192 provider may consider stopping the sequence

2193

2194 **Synopsis:** PUT {URI of the scalability group}

2195 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
2196 Authorization, Accept

2197 **Request Message Body:** ScalabilityGroup2198 **Response Headers:** Content-Length, Content-Type, Content-Location2199 **Response Message Body:** ScalabilityGroup2200 **Response Status:** 200, 202, 400, 401, 403, 404, 409

2201

2202 Example Request: shutting down a scalability group and response with resource_state field only

```

2203 PUT /123/vdcs/vdc232/c54
2204 Host: cloudcompany.com
2205 Authorization: Basic xxxxxxxxxxxx
2206 Content-Length: nnn
2207 Content-Type:
2208 application/vnd.com.oracle.cloud.ScalabilityGroup+json
2209 Accept:
2210 application/vnd.com.oracle.cloud.ScalabilityGroup+json?resourcest*
2211 X-Cloud-Client-Specification-Version: 0.1
2212 {
2213   "status" : "OFFLINE"
2214 }

```

2215

2216 Example Response:

```

2217 HTTP/1.1 202 Accepted
2218 Content-Type:
2219 application/vnd.com.oracle.cloud.ScalabilityGroup+json
2220 Content-Location: /123/vdcs/vdc232/c54
2221 Content-Length: nnn
2222 {
2223   "uri" : "/123/vdcs/vdc232/c54",
2224   "resource state" : {
2225     "state" : "SHUTTING DOWN",
2226     "progress" : "40",
2227     "message" : [
2228       {"code" : "1", "text" : "shutting down node 1"},
2229       {"code" : "1", "text" : "shutting down node 2"},
2230       ...
2231     ]
2232   }
2233 }

```

2234

2235 18.4 Scale the ScalabilityGroup

2236 To scale out or scale in a scalability group, the user can do a PUT of the desired *count* to the scalability
 2237 group resource overwriting the current *count* of the scalability group. The server SHOULD ensure the
 2238 “same-ness” by observing what other existing nodes in the scalability group are and then create additional
 2239 resource into the scalability group automatically.

2240

2241 A service provider may consider the following:

- 2242 ▪ If a scalability group is already being scaled out to a number greater than the request,
 2243 the server can return 202
- 2244 ▪ If a scalability group is being scaled in to a number greater than the request, the server
 2245 can return 202 and further scaled in to the smaller number
- 2246 ▪ If a scalability group is being requested to change size by two requests that are obviously
 2247 in opposite direction, then the server may reject either one or both requests. For
 2248 example, a scalability group with 4 nodes where request A POST to go to 1 node, and
 2249 then followed by request B POST to go to 7 node. The server may return 409 on
 2250 request B

2251

2252 **Synopsis:** PUT {URI of the scalability group}

2253 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 2254 Authorization, Accept

2255 **Request Message Body:** ScalabilityGroup

2256 **Response Headers:** Content-Length, Content-Type, Content-Location

2257 **Response Message Body:** ScalabilityGroup

2258 **Response Status:** 200, 202, 400, 401, 403, 404, 409

2259

2260 Example Request: scale out a scalability group from 5 to 8

```

2261 PUT /123/vdcs/vdc232/c54
2262 Host: cloudcompany.com
2263 Authorization: Basic xxxxxxxxxxxx
2264 Content-Length: nnn
2265 Content-Type:
2266 application/vnd.com.oracle.cloud.ScalabilityGroup+json
2267 Accept:
2268 application/vnd.com.oracle.cloud.ScalabilityGroup+json?resourcest*
2269 X-Cloud-Client-Specification-Version: 0.1
2270 {
2271   "count" : "8"
2272 }
  
```

2273

2274 Example Response:

```

2275 HTTP/1.1 202 Accepted
2276 Content-Type:
2277 application/vnd.com.oracle.cloud.ScalabilityGroup+json
2278 Content-Location: /123/vdcs/vdc232/c54
2279 Content-Length: nnn
2280 {
2281   "uri" : "/123/vdcs/vdc232/c54",
  
```

```

2282     "resource state" : {
2283         "state" : "SCALING OUT",
2284         "progress" : "80",
2285         "message" : [
2286             { "code" : "1", "text" : "preparing resources" },
2287             { "code" : "1", "text" : "adding nodes to cluster" },
2288             ...
2289         ]
2290     }
2291 }

```

2292

2293

2294 [Return to Section List](#)

2295 19 Operations on the ServiceTemplate resource

2296 Operations on a ServiceTemplate resource allow the client to

- 2297 ■ view the attributes of the service template, and for users, the definition of the service
- 2298 template

2299 19.1 Retrieve information about the service template.

2300

2301 **Synopsis:** GET {URI of a Service template instance}2302 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization2303 **Request Message Body:** N/A.2304 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location2305 **Response Message Body:** ServiceTemplate2306 **Response Status:** 200, 400, 401, 403, 404

2307

2308 Example Request: Retrieve the information about a service template

```

2309 GET /templates/items/t833
2310 Host: cloudcompany.com
2311 Authorization: Basic xxxxxxxxxxxx
2312 Accept: application/vnd.com.oracle.cloud.ServiceTemplate+json
2313 X-Cloud-Client-Specification-Version: 0.1

```

2314

2315 Example Response

```

2316 HTTP/1.1 200 OK
2317 Content-Type: application/vnd.com.oracle.cloud.ServiceTemplate+json
2318 Content-Location: /templates/items/t833
2319 Content-Length: nnn
2320 {
2321     "uri" : "...",
2322     "name" : "Oracle EM NextGen Demo in a box",
2323     "description" : "Template that can be deployed into complete EM
2324     Next Generation Demo system with..." ,
2325     "type" : "VMTemplate" ,

```



```

2326     "created" : "February 22, 2010 15:34:34 GMT" ,
2327     "definition" : "<xml><some definition/>...</xml>" ,
2328     "tags" : ["ABC", "EM", "Next Gen", "My note", ... ] ,
2329     "resource state" : [
2330         "state" : "READY"
2331     ]
2332 }

```

2333

2334 19.2 Deletes a service template

2335 The cloud service provider may allow users to delete the service templates, based on the provider specified
 2336 accessibility rules. However, since there may be active instances referencing to the service template, the
 2337 cloud service provider may reserve the right to remove only the visibility of the service template from any
 2338 menus. In that case, subsequent GET request to a deleted service template may still be valid.

2339

2340 **Synopsis:** DELETE {URI of a service template}

2341 **Request Headers:** Host, X-Cloud-Client-Specification-Version, Authorization

2342 **Request Message Body:** N/A

2343 **Response Headers:** N/A

2344 **Response Message Body:** N/A

2345 **Response Status:** 200, 401, 403, 404, 410

2346

2347 **Example Request:** Delete a service template

```

2348 DELETE /templates/items/t903
2349 Host: cloudcompany.com
2350 Authorization: Basic xxxxxxxxxxxx
2351 X-Cloud-Client-Specification-Version: 0.1

```

2352

2353 **Example Response**

```

2354 HTTP/1.1 200 OK

```

2355

2356 [Return to Section List](#)

2357

2358 20 Operations on AssemblyInstance resources

2359 Requests to AssemblyInstance allow the user to

- 2360
 - View the Assembly Instance information, including all of its direct sub components
 - 2361
 - Create a new Assembly Instance
 - 2362
 - By posting to *cloud.vdcs.AssemblyInstances.uri*

2363

2364 A Assembly Instance can be instantiated via

2365 ▪ based_on a ServiceTemplate reference

2366

2367

2368 **20.1 Get Assembly Instance**

2369 Retrieving the details of a Assembly Instance.

2370

2371 **Synopsis:** GET {URI of a Assembly Instance}

2372 **Request Headers:** Host, Accept, X-Cloud-Client-Specification-Version, Authorization

2373 **Request Message Body:** N/A.

2374 **Response Headers:** Content-Length, Content-Type, Cache-Control, Content-Location

2375 **Response Message Body:** AssemblyInstance

2376 **Response Status:** 200, 400, 401, 403, 404, 410

2377

2378 Example Request: Retrieve the AssemblyInstance attributes

```
2379 GET /123/dg/34
2380 Host: cloudcompany.com
2381 Authorization: Basic xxxxxxxxxxxx
2382 Accept: application/vnd.com.oracle.cloud.AssemblyInstance+json
2383 X-Cloud-Client-Specification-Version: 0.1
```

2384

2385 Example Response

```
2386 HTTP/1.1 200 OK
2387 Content-Type:
2388 application/vnd.com.oracle.cloud.AssemblyInstance+json
2389 Content-Location: /123/dg/34
2390 Cache-Control: no-store
2391 Content-Length: nnn
2392 {
2393   "uri" : "...",
2394   "name" : "Testing Site: Enterprise Manager with RAC Database
2395 11.2",
2396   "description" : "This is an instantiation of Oracle Enterprise
2397 Manager 11.2 with RAC database along with multiple agents for the
2398 standard QA testing and regressions..." ,
2399   "tags" : ["Testing", "QA", "EM", "RAC", ... ] ,
2400   "based on" : "/templates/items/t835" ,
2401   "scalability_groups" : {
2402     "uri" : "/123/dg/34/scalability groups",
2403     "type" : "ScalabilityGroup",
2404     "total" : "2" ,
2405     "elements" : [
2406       {"uri": "/123/dg/34/c24", "name": "RAC Database"},
2407       {"uri": "/123/dg/34/c43", "name": "FMW Application Server
2408 - OMS"}
2409     ]
2410   } ,
2411   "servers" : {
```

```

2412     "uri" : "/123/dg/34/server" ,
2413     "type" : "Server" ,
2414     "total" : "3" ,
2415     "elements" : [
2416         { "uri" : "/123/dg/34/server23", "name" : "Website Load
2417 Tester" } ,
2418         { "uri" : "/123/dg/34/vm34", "name" : "Simple Web Server
2419 for serving Reports" } ,
2420         { "uri" : "/123/dg/34/server49", "name" : "Database to
2421 store testing results" }
2422     ]
2423 } ,
2424 "status" : "ONLINE" ,
2425 "zone" : "/123/wczone" ,
2426 "resource state" : {
2427     "state" : "READY"
2428 } ,
2429 "created" : "February 26, 2010 15:34:34 GMT" ,
2430 "expiry" : "March 26, 2010 15:34:34 GMT" ,
2431 "contains in" : "/123/vdcs/vdc232"
2432 }

```

2433

2434 20.2 Creates a new Assembly Instance

2435 In this example, the user wants to add a new Peoplesoft system to the VDC.

2436

2437 **Synopsis:** POST {URI of the VDC}

2438 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
2439 Authorization, Accept

2440 **Request Message Body:** AssemblyInstance

2441 **Response Headers:** Content-Length, Content-Type, Content-Location

2442 **Response Message Body:** AssemblyInstance

2443 **Response Status:** 200, 202, 400, 401, 403, 404, 409

2444

2445 Example Request: creates a new Assembly Instance under an existing Assembly Instance

```

2446 POST /123/vdcs/vdc232
2447 Host: cloudcompany.com
2448 Authorization: Basic xxxxxxxxxxxx
2449 Content-Length: nnn
2450 Content-Type:
2451 application/vnd.com.oracle.cloud.AssemblyInstance+json
2452 Accept: application/vnd.com.oracle.cloud.AssemblyInstance+json
2453 X-Cloud-Client-Specification-Version: 0.1
2454 {
2455     "name" : "Peoplesoft System - QA1" ,
2456     "based on" : "/templates/items/t943",
2457     "zone" : "/123/wczone" ,
2458     "params" : {
2459         "em agent install" : "TRUE",
2460         "em_oms" : "https://192.168.4.203:7844/upload",
2461         "em secured key" : "k3sxxkjs92kx0lka-1ks02j01jd02",
2462         ...

```

2463 }
 2464 }

2465

2466 Example Response:

```

2467 HTTP/1.1 202 Accepted
2468 Content-Type:
2469 application/vnd.com.oracle.cloud.AssemblyInstance+json
2470 Content-Location: /123/dg/103
2471 Content-Length: nnn
2472 {
2473   "uri" : "/123/dg/103",
2474   "name" : "Peoplesoft System - QA1",
2475   "resource state" : {
2476     "state" : "CREATING",
2477     "progress" : "10",
2478     "message" : [
2479       { "code" : "2", "text" : "validating parameters" },
2480       { "code" : "6", "text" : "preparing images" },
2481       ...
2482     ]
2483   }
2484 }
```

2485

2486 Since the resource state is not in READY state, the client should poll /123/dg/103?resource_state to
 2487 periodically check whether or not the Assembly Instance creation is completed.

2488

2489 20.3 Shutdown the Assembly Instance

2490 To shutdown the Assembly Instance, the user can PUT the OFFLINE status to the Assembly Instance.

2491

2492 A service provider may consider the following:

- 2493 ▪ If a Assembly Instance is depended upon by another entity, shutting down the Assembly
 2494 Instance may result in an 421 (Dependency Not Allowed) error
- 2495 ▪ If a Assembly Instance is starting up when a shutdown request is submitted, the service
 2496 provider may consider stopping the sequence

2497

2498 **Synopsis:** PUT {URI of the Assembly Instance}

2499 **Request Headers:** Host, Content-Length, Content-Type, X-Cloud-Client-Specification-Version,
 2500 Authorization, Accept

2501 **Request Message Body:** AssemblyInstance

2502 **Response Headers:** Content-Length, Content-Type, Content-Location

2503 **Response Message Body:** AssemblyInstance

2504 **Response Status:** 200, 202, 400, 401, 403, 404, 409

2505

2506 Example Request: shutting down a Assembly Instance and response with resource_state field only

```
2507 PUT /123/dg/34
2508 Host: cloudcompany.com
2509 Authorization: Basic xxxxxxxxxxxx
2510 Content-Length: nnn
2511 Content-Type:
2512 application/vnd.com.oracle.cloud.AssemblyInstance+json
2513 Accept:
2514 application/vnd.com.oracle.cloud.AssemblyInstance+json?resourcec*
2515 X-Cloud-Client-Specification-Version: 0.1
2516 {
2517   "status" : "OFFLINE"
2518 }
```

2519

2520 Example Response:

```
2521 HTTP/1.1 202 Accepted
2522 Content-Type:
2523 application/vnd.com.oracle.cloud.AssemblyInstance+json
2524 Content-Location: /123/dg/34
2525 Content-Length: nnn
2526 {
2527   "uri" : "/123/dg/34",
2528   "resource state" : {
2529     "state" : "SHUTTING DOWN",
2530     "progress" : "10",
2531     "message" : [
2532       { "code" : "1", "text" : "shutting down web server" },
2533       { "code" : "1", "text" : "shutting down database" },
2534       ...
2535     ]
2536   }
2537 }
```

2538

2539

2540 [Return to Section List](#)

2541

**ANNEX A
(informative)**

Change Log

2542
2543
2544
2545

Change Log		
Version	Author	Changes
1.0/0.33	jack.yu	Initial Creation of the Oracle Cloud Resource Model API
1.0/0.34	jack.yu	Tighten the definition of Assembly Instance into Assembly Instance

2546
2547