

ORACLE®

Cloud Computing: IaaS Cloud Resource Models

Oracle Cloud Resource Model API

- Models fundamental resources offered in the Oracle Cloud solutions
- Leverages the work done on Sun Cloud model/APIs
 - Uses HTTP methods to interact with the resources to achieve provisioning, association, modifying, and retiring of entities
 - Adheres to REST architecture style with JSON style payload
- Oracle Elemental Resource Model
 - Focusing on IaaS
 - Contains top level, non-assuming, aggregation resources
 - Zone
 - VDC



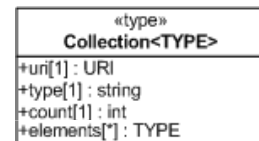
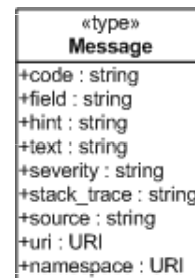
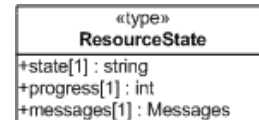
REST architecture on HTTP + JSON

- Why REST architecture?
 - interfacing entities are resources
 - hyperlink-driven meshes well with entity relationships
- Why HTTP protocol?
 - operations (PUT/POST/DELETE/GET/...) are concise and simple
 - work through firewalls, proxies, and caches
 - horizontal scalable (front end load balancer)
- Why JSON?
 - work with JavaScript in browser interfaces
 - human readable
 - minimal encoding overhead
 - extensible with best effort validation
- Common tools and libraries for programming languages are becoming available for REST/HTTP + JSON
- Separation of protocol and resources holds well for “stacking-up” with multi-layer compatibility



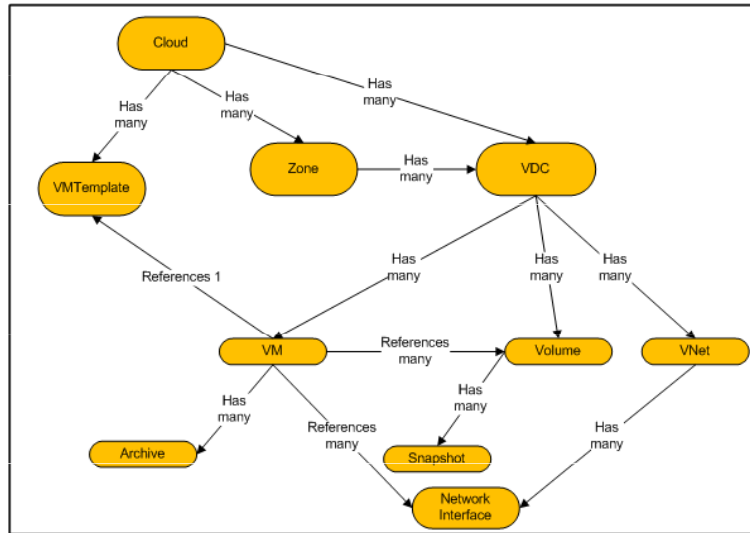
Common Behaviors

- Each resource contains ResourceState attribute
 - to signify whether or not the resource itself is ready to be consumed
 - this is semantically different from the “status” of the entity represented by the resource and provide additional information than HTTP status code
- Responses with Error Codes (4xx & 5xx) should accompany with Messages resource
 - contains a list of messages that contain hints and stack-trace (where appropriate)
 - each Message contains a namespace for standard message space
- Collection Data Model to enhance the generic List
 - type strong
 - each collection has its own uri attribute
 - contains the auxiliary count attribute
- There is no explicit operational Status resource
 - a resource’s ResourceState indicates the availability on the same uri





Cloud Resource Model - Elemental



Cloud

- Entry point to the resources, and a published link
 - Identity, accessibility, and privileges would be handled implicitly as soon as a client establishes credential
 - references
 - 1 or more VDCs
 - 1 or more Zones
 - 0 or more VM Templates
- This resource contains *profiles* attribute that describes characteristics of the underlying infrastructures
 - The elements in the profiles can be used to examine zones and identify which may satisfy the service characteristics. For example, “HIPPA compliance” is a service profile that maybe a subset of zones supports

Cloud
+uri[1] : URI
+specification_version[*] : string
+implementation_version[1] : string
+name[1] : string
+description[1] : string
+tags[*] : string
+service_templates[1] : Collection<VMTemplate>
+zones[1] : Collection<Zone>
+vdc[1] : Collection<VDC>
+resource_state[1] : ResourceState
+profiles[1]





VDC

- This is short for Virtual Data Center, and is a grouping of client's resources that could be constrained on the underlying physical infrastructure
 - Identity, accessibility, and privileges would be handled implicitly universally as soon as a client establishes credential
 - references
 - 1 Zone – all resources in the VDC are collocated in the zone
 - contains
 - 0 or more VMs
 - 0 or more Volumes
 - 0 or more VNets

VDC
+uri[1] : URI
+name[1] : string
+description[1] : string
+tags[*] : string
+scalability_groups[1] : Collection<ScalabilityGroup>
+servers[*] : Collection<VM>
+volumes[1] : Collection<Volume>
+vnets[1] : Collection<VNet>
+zone[1] : URI
+resource_state[1] : ResourceState
+created[1] : DateTimeUTC
+params[1]



Zone

- This represents a boundary of which resource collocation implies service characteristics. For example,
 - a Zone with profile LAN could mean that resources collocated in the zone could expect LAN latency
 - a Zone with profile DEV could mean that resources collocated in the zone could expect development environment QoS
- This resource contains *profiles* attribute that contains descriptions on how the zone may be organized and what kind of services it offers
- Multiple VDCs can reside in a single Zone

Zone
+uri[1] : URI
+name[1] : string
+description[1] : string
+tags[*] : string
+profiles[*]
+platform[1] : string





VM Template

- This is a deployable entity that can be realized into a VM
 - *params* may contain configuration metadata

VMTemplate
+uri[1] : URI
+name[1] : string
+description[1] : string
+created[1] : DateTimeUTC
+params[*]
+based_on[1] : Archive
+tags[*] : string
+resource_state[1] : ResourceState
+os[1] : string
+cpu[1]
+memory[1] : int
+disks[1]



VM

- This is a computing container providing a OS-based system platform, and represents a virtual machine
 - references
 - 1 VDC
 - 0 or 1 VM Template
 - contains
 - 0 or more Volumes
 - 0 or more Network Instances
 - 0 or more Archives
- Archive is the backup of a VM

VM
+uri[1] : URI
+name[1] : string
+description[1] : string
+tags[*] : string
+contained_in[1] : URI
+container_type[1] : string
+status[1] : string
+based_on[1] : URI
+hostname[1] : string
+cpu[1]
+memory[1] : int
+disks[1]
+volumes[1] : Collection<Volume>
+interfaces[1] : Collection<NetworkInterface>
+params[1]
+archives[1] : Collection<Archive>
+clone_from[1] : URI
+resource_state[1] : ResourceState
+created[1] : DateTimeUTC
+type[1] : string
+restored_from[1] : URI

Archive
+uri[1] : URI
+name[1] : string
+description[1] : string
+created[1] : DateTimeUTC
+source[1] : URI
+resource_state[1] : ResourceState
+params[1]





Volume

- Represents a storage that may exist independently, and thus differentiate it from the “local disks” of a VM
- All volumes are networked
- A volume may be sharable among different VMs given the underlying infrastructure constraints are satisfied
 - Contains
 - 0 or more Snapshots
- Snapshot is the backup of a volume

Volume
+uri[1] : URI
+name[1] : string
+description[1] : string
+tags[*] : string
+contained_in[1] : URI
+container_type[1] : string
+params[1]
+size[1] : int
+snapshots[1] : Collection<Snapshot>
+clone_from[1] : URI
+created[1] : DateTimeUTC
+resource_state[1] : ResourceState
+restored_from[1] : URI

Snapshot
+uri[1] : URI
+name[1] : string
+description[1] : string
+created[1] : DateTimeUTC
+source[1] : URI
+resource_state[1] : ResourceState
+params[i]



VNet

- A service that is capable of providing network addressing, routing rules, security constraints, and access limits
- Contains
 - 0 or more Interfaces
- “base_network” field can be used to suggest connectivity between VNets
- VNet can be extended to specify services such as load balancer, DMZ and others. This specification does not enumerate them, nor does it assume any underlying network topologies

VNet
+uri[1] : URI
+name[1] : string
+description[1] : string
+tags[*] : string
+contained_in[1] : URI
+container_type[1] : string
+params[1]
+created[1] : DateTimeUTC
+interfaces[1] : Collection<NetworkInterface>
+based_on[1] : URI
+base_network[1] : URI





NetworkInterface

- An instance of a network endpoint consists of complete address that can be interpreted by the underlying network infrastructure
- References
 - 1 VNet

NetworkInterface
+uri[1] : URI
+name[1] : string
+description[1] : string
-routable[1] : bool
+vnet[1] : URI
+address[1] : string
+address_type[1] : string



Extensibility of the model

- “tags” field of resources can be used to support ad-hoc resource attributes before standardized
- addition fields could be added
 - field name must adhere to the character limits for GET request parameters supports
 - for POST-able and PUT-able fields, defaults must be implemented to support baseline client implementations
- message code may extend the namespaces