Agile Lifecycle Manager

Resource Descriptor Specification

Version: 1.0.0

Disclaimer Note

Document History

Date	Change Description	Author
26/09/2017	Initial version	Martin Roberts

Table of Contents

1. Introduction

This document describes the descriptors that are used by the Agile Lifecycle Manager (ALM). The ALM needs to have descriptions of the building blocks of applications that it is going to manage. The basic building blocks are described in resource descriptors. Sets of these are composed into assembly descriptors to allow designers to describe a complete application/service that they need the ALM to manage.

1.1. Naming

The resource descriptor name field will contain the following string:

```
        resource::name::1.0
```

The name must start with a letter (either case) and can include letter and numbers and underscore and hyphens (minus sign). The name must not contain spaces.

The version is fixed to 1.0 for this release.

Both name and version are mandatory.

2. Descriptor Sections

The following sections of the descriptors can occur in resource descriptors.

2.1. Header

The header of a resource descriptor includes the name and description of the descriptor and associated resource manager type. Each resource is associated with a resource manager that has a declared type. This is shown by the field resource-manager-type. The contents of the field must be a globally unique string.

```
name: resource::c_streamer::1.0
description: component package for  c_streamer
resource-manager-type: urbancode.ibm.com
```

2.2. ‘properties’

(The properties section occurs in a number of places. The rules defined here apply to all property sections)

This section contains the properties that belong to the resource descriptors. These include the full set of properties that are required to orchestrate them through to the Active state. These can be understood as the context for the management of the item during its lifecycle.

```
properties:
  deploymentLocation: # the name of the property
    type: string
    required: true
    description: The name of the openstack project(tenant) to install this assembly in.
  numOfStreamers:
    type: string
    description: the number of streamers that should be created at install time
    default: 2
  tenant_key_name:
    type: string
    required: true
    description: The ssh key for the current tenant
  flavor:
    value: m1.small
  cluster_public_ip_address:
    type: string
    description: the public IP address for this cluster
    read-only: true
    value: '${balancer.publicIp}'
```

Each property name must be unique within its property section. The types of properties can be string. Password indicated fields that will contain passwords or sensitive data. Properties are optional unless explicitly defined as required by the inclusion of a `required: true` flag.

Properties marked as `read-only: true` will typically have that value set by the time the associated component instance is in the Active state. These fields must not be marked as `required: true`.

Properties may be declared with a `default` value or a specific `value` or neither. Where the value field is used it may either be an explicit value or it may reference to another property within the description. When referencing a property reference will look as follows: `value: '${max_connections}'`

The ALM will assign an internal name and identifier for each resource instance it creates. These values can be useful to give unique names for servers etc. To access them a property may have its value set to ${instance.name} or ${instance.id}.

2.3. Capabilities and Requirements

These two sections allow designers to explain what functions the resources are implementing or need before they can work successfully. These might be expressing that networks or various types must be available for the resource instances to work or it may be describing that a resource supports, for example, incoming http requests.

The type is a string that expresses the capability or requirement. The values in these string will have to be agreed across an organisation and where possible they should be agreed by the industry. Resource capabilities should use common industry terms. In the examples below the idea is that httpStreamOutput indicates that the capability is using the http protocol in a stream form and in an output direction. The OS::Neutron:Net is the resource type from openstack associated with a network instantiated within neutron.

2.3.1. 'capabilities'

These are used to enable service designers to understand what function a resource provides.

```
capabilities:
    VideoStream:
        type: httpStreamOutput
```

```
capabilities:
    Network:
        type: neutronNetwork
```

2.3.2. 'requirements'

Similar to the capabilities the requirements contain the list of capabilities that the resource needs to be provided for them to work.

```
requirements:
    VideoNetwork:
        type: neutronNetwork
    ManagementNetwork:
        type: neutronNetwork
    RemoteNFSMountPoint:
        type: nfsExportMountpoint
```

2.4. ‘operations’

This section defines sets operations that can be called to enable relationships to be created between resources. Operations definitions in the resource have a name and a set of properties. Where a resource descriptor describes an operation. As a convention the name of the operation should be linked to the capability that is being enabled through the creation of the relationship.

```
operations:
  RemoveHttpStreamOutput:
    description: removes the http server from being managed by the balancer
    properties:
      server_ip:
        type: string
        description: Http Server Ip Address
        default: the ip address
      server_port:
        type: string
        description: http server port number
        default: '8080'
  AddHttpStreamOutput:
    description: adds an http server to the balancer's pool
    properties:
      max_connections:
        type: string
        description: Maximum connections for the balanced server
        default: 3
      server_ip:
        type: string
        description: Ip Address of the server to be balanced
      server_port:
        type: string
        description: Port on balanced server
        default: '8080'
```

Resource Descriptor Operations

2.5. ‘lifecycle’

Resource descriptors must support the Install and Uninstall lifecycle transitions. These are mandatory. However, they may implement the other lifecycle transitions which are: Configure, Start, Stop, and Integrity.

Where the transition is not provided by the resource the ALM will be free to change the state of the associated component instances without calling any underlying transition.

The lifecycle section will contain a list of all the transitions that the resource supports.

```
lifecycle:
- Install
- Uninstall
- Start
- Stop
- Integrity
```

Notice that in this example there is no Configure transition defined in this example.

A resource may be one that can only be used within a reference section of an assembly. These resources will not have an Install or Uninstall lifecycle defined.

Resources that are used as reference resources do not have to include the ‘lifecycle’ section. Any resource without the Install and Uninstall cannot be instantiated by the lifecycle manager and therefore should not be included in the composition section of an assembly

3. Yaml Examples

The examples included below show the c_balancer,c_streamer and the net_video resources.

3.1. 'resource' examples

3.1.1. resource::net_video:1.0

This is a resource that creates a neutron network

```
name: resource::net_video::1.0
description: resource to create an internal neutron network that includes a subnet
resource-manager-type: urbancode.ibm.com
properties:
  subnetCIDR:
    type: string
    description: The subnet classless inter-domain routing
    default: '10.0.1.0/24'
  networkName:
    type: string
    description: Network Name
    value: VIDEO
  subnetDefGwIp:
    type: string
    description: Default Gateway IP address
    default: '10.0.1.1'
  network-id:
    type: string
    description: the id of the network just created
    read-only: true
capabilities:
    Network:
        type: OS::Neutron::Net
lifecycle:
- Install
- Uninstall
```

3.1.2. A simple component with metrics and policies

```
name: "resource::h_simple::1.0"
description: "resource for  t_simple"
properties:
  server_name:
    type: "string"
    value: "${instance.name}"
  referenced-internal-network:
    type: "string"
    description: "Generated to reference a network"
  reference-public-network:
    type: "string"
    description: "Generated to reference public network"
  image:
    type: "string"
    description: "The Image reference"
  key_name:
    type: "string"
    description: "SSH key"
  data:
    type: "string"
    description: "parameter passed"
    default: "data"
  integrity_publication_period:
    type: "string"
    description: "the period that should be used to publish the metrics"
    default: "60"
  publication_period:
    type: "string"
    description: "the period that should be used to publish the metrics"
    default: "60"
  number-of-intervals:
    type: "string"
    description: "The intervals before calling a Heal"
    default: "3"
  output:
    type: "string"
    description: "an example output parameter"
    read-only: true
operations:
  CreateRelationship1:
    description: "Create a new relationship"
    properties:
      source:
        type: "string"
        description: "that name of the source"
      target:
        type: "string"
        description: "that name of the target"
  CeaseRelationship1:
    description: "Cease an existing relationship"
    properties:
      source:
        type: "string"
        description: "that name of the source"
      target:
        type: "string"
        description: "that name of the target"
  CreateRelationshipr2:
    description: "Create a new relationship"
    properties:
      source:
        type: "string"
        description: "that name of the source"
      target:
        type: "string"
        description: "that name of the target"
  CeaseRelationship2:
    description: "Cease an existing relationship"
    properties:
      source:
        type: "string"
        description: "that name of the source"
      target:
        type: "string"
        description: "that name of the target"
  CreateRelationship3:
    description: "Create a new relationship"
    properties:
      source:
        type: "string"
        description: "that name of the source"
      target:
        type: "string"
        description: "that name of the target"
  CeaseRelationship3:
    description: "Cease an existing relationship"
    properties:
      source:
        type: "string"
        description: "that name of the source"
      target:
        type: "string"
        description: "that name of the target"
lifecycle:
- "Configure"
- "Install"
- "Integrity"
- "Start"
- "Stop"
- "Uninstall"
resource-manager-type: "test-rm"
```

3.1.3. resource::c_streamer::1.0

This descriptor will create a virtual server that streams video traffic using the http protocol.

```
name: resource::c_streamer::1.0
description: resource descriptor for  c_streamer
resource-manager-type: urbancode.ibm.com
properties:
  key_name:
    type: string
    required: true
    description: the ssh key-pair name to be used by openstack with the associated VM instances
  referenced-management-network:
    type: string
    required: true
    description: The id of the network that will act in the role of a management network
  flavor:
    type: string
    required: true
    description: Flavor to be used for compute instance
  server_name:
    type: string
    required: true
    description: the name of the server to be created
  referenced-video-network:
    type: string
    description: The id of the network that will act in the role of an internal network
  availability_zone:
    type: string
    description: Name of availability zone in which to create the instance
    default: DMZ
  privateIp:
    type: string
    description: IpAddress of server on the internal network
    read-only: true
  mgmtIp:
    type: string
    description: IpAddress of server on the management network
    read-only: true
  integrity_publication_period:
    type: string
    description: the number of seconds between publishing integrity metric
    default: 60
  number-of-intervals:
    type: string
    description: the number of intervals for smoothing
    default: 3
capabilities:
    VideoStream:
        type: httpStreamOutput
requirements:
    VideoNetwork:
        type: neutronNetwork
    ManagementNetwork:
        type: neutronNetwork    RemoteNFSMountPoint:
        type: nfsExportMountpoint
lifecycle:
- Install
- Uninstall
- Configure
- Start
- Stop
- Integrity
operations:
  MountStorage:
    description: An operation to enable the streamer to mount a remote NFS mount point
    properties:
      remote_nfs_port:
        type: string
        description: Port for the NFS
        default: '2049'
      remote_nfs_server_ip:
        type: string
        description: Ip Address of remote nfs server
      remote_mount_point:
        type: string
        description: Location of NFS Exported Mount Point
        default: /
      local_mount_point:
        type: string
        description: The location where the remote nfs mount will be mounted in the local machine
        default: /mnt
  UnmountStorage:
    description: An operation to unmount a remote NFS mount point
    properties:
      local_mount_point:
        type: string
        description: The location where the remote nfs mount will be mounted in the local machine
        default: /mnt
```

3.1.4. resource::c_balancer::1.0

```
name: resource::c_balancer::1.0
description: component package for a http loadbalancer
resource-manager-type: UrbanCode
cloud-target: OpenStack
properties:
  key_name:
    type: string
    description: ssh key_name.
  referenced-management-network:
    type: string
    description: Generated to reference a network
  referenced-internal-network:
    type: string
    description: Generated to reference a network
  referenced-public-network:
    type: string
    description: Generated to reference a network
  flavor:
    type: string
    description: Flavor to be used for compute instance
  server_name:
    type: string
    description: server name of the balancer
  availability_zone:
    type: string
    description: Name of availability zone in which to create the instance
    default: DMZ
  mgmtIp:
    type: string
    description: IpAddress of server in management network
    readOnly: true
  internalIp:
    type: string
    description: IpAddress of server on internal network
    readOnly: true
  publicIp:
    type: string
    description: Public IpAddress of server
    readOnly: true
  integrity_publication_period:
    type: string
    description: the number of seconds between publishing integrity metric
    default: 60
  number-of-intervals:
    type: string
    description: the number of intervals for smoothing
    default: 3
capabilities:
    HttpLoadBalancer:
        type: loadbalancerHttp
requirements:
    PublicNetwork:
        type: neutronNetwork
    ManagementNetwork:
        type: neutronNetwork
    HttpServer:
        type: http
lifecycle:
- Install
- Uninstall
- Start
- Stop
operations:
  RemoveBalancedHttpServer:
    description: removes the http server from being managed by the balancer
    properties:
      server_ip:
        type: string
        description: Http Server Ip Address
        default: the ip address
      server_port:
        type: string
        description: http server port number
        default: '8080'
  AddBalancedHttpServer:
    description: adds an http server to the balancer's pool
    properties:
      max_connections:
        type: string
        description: Maximum connections for the balanced server
        default: 3
      server_ip:
        type: string
        description: Ip Address of the server to be balanced
      server_port:
        type: string
        description: Port on balanced server
        default: '8080'
```