**Resource Group API**

To consume the resource group API a token must be generated first.

To create a token a request can be made to the /api/v2/cdos/tokens

Example:

**Request:**

url: <http://cdos.dev.cloud/api/v2/cdos/tokens>

method: post

body:

username: user

password: password

**Response:**

body: token:

tokenId: 3emas9NQAQJXY5UuUEKUqWpU\_nb0bWnMdoSxjYNIIJ8xixnY9jnT3XqljT0Yx0ujBeEe9EARN\_y\_F3VK6F8oAA

expire: 28-Jan-14 06:43:06

created: 28-Jan-14 06:38:06

user:

id: 67

username: kam

email: kam@ivedha.com

firstname: Kam

lastname: Patel

enabled: false

accessLevel: CLOUD

accessLevelType: CLOUD

functionalRole: Site Admin

Once the token has been generated it can be used in the following requests to authenticate the user.

With the token in hand the resource group can then be created.

There are 4 steps involved the group created, bellow is a description of each one individually:

\*One important point to note is that the RESOURCE\_ID attribute must be used when interacting with resource group API

#1 Select a template

**Request:**

url: http://cdos.dev.cloud/api/v2/cdos/offerings?type=TEMPLATE

json: true

headers:

X-Auth-Token: 4fI4sy9QBsqV4qKEdKd6nNTgK8HovgwtReEYivyybwodxy\_QanzGUT8\_dS7KOrqtvsgXFqaViPkEQkkCjLRktA

**Response:**

offerings:

-

id: a00842a1-3705-4f0c-bb71-52ddc7488c60

**resourceId: 59a4e7da-348e-4e71-8838-edd800c52818 -> Keep track of this ID**

name: MasterPuppet

resourceState: Online

siteId: 1

siteName: Earth42

accountId: 67

cru: 0

createdTime: 2014-01-22T16:23:29-0500

hypervisor: XenServer

osTypeId: 1cb3def6-35d1-11e3-a6fd-3a96760eadba

osTypeName: CentOS 6.2 (64-bit)

format: VHD

state: Download Complete

size: 20

publicAccess: true

featured: true

extractable: true

passwordEnabled: false

type: featured

ready: true

storageCru: 0.12000000104308128

templateType: TEMPLATE

-

#2 Select a compute offering

**Request:**

url: http://cdos.dev.cloud/api/v2/cdos/offerings?type=COMPUTE

json: true

headers:

X-Auth-Token: VVUgYz9JK17kLJhEQgscyN32RoMMfUbANO2tVc1klsw2oOsYrzhBF04bKu-s4PuyssTUmQv0rUPZ0vbxp0zeQw

**Response:**

offerings:

-

id: 3245

siteid: 1

accountid: 1

siteresourceid: c8f39d44-be8e-46c7-b754-f903e5ad4dcc

offeringType: COMPUTE

cru: 0

created: 1388717071000

state: ONLINE

name: rfs.small

**resourceid: 6f0bc41a-883e-11e3-9605-be159b62ffc0 -> Keep track of this ID**

userid: 1

#3 Select a security zone

**Request:**

url: <http://cdos.dev.cloud/api/v2/cdos/securityzones>

json: true

headers:

X-Auth-Token: 0rU7PVWmiSPF\_vHz7nuh9zuKEjW4iLUv2pxib7ZPh4ZJU-AytsK9H6uNl3pp9y64b\_rcTUZh8IHJP\_WxsWiAbA

**Response:**

securityZones:

-

id: 239b4055-b155-4c9c-aa4f-fd8447d73fdf

**resourceId: a91e23fe-da74-4a35-8fe3-93bfaaef95f0 -> Keep track of this ID**

name: Web

resourceState: Online

siteId: 1

siteName: Earth42

accountId: 67

cru: 0

desc: Web Zone

type: basic

#4 Create Resource Group

**Request:**

url: http://cdos.dev.cloud/api/v2/cdos/resourcegroups

json: true

headers:

X-Auth-Token: x9GaKvSQk1tB5Hcny0s6fVXtcY4OqWwTQiqYw6CWwfxwwrVtEue25HclhRIe3otyyhtys84\_UpN014vqtf\_38g

body:

siteId: 1

computeResourceId: **6f0bc672-883e-11e3-9605-be159b62ffc0**

templateResourceId: **59a4e7da-348e-4e71-8838-edd800c5281**

securityResourceId: **a91e23fe-da74-4a35-8fe3-93bfaaef95f0**

name: bidder1101

**Response:**

resourcegroup:

id: 9

siteId: 1

accountId: 67

computeResourceId: 6f0bc672-883e-11e3-9605-be159b62ffc0

templateResourceId: eb0902d5-e1dd-4540-9468-230a0c44c589

securityResourceId: a91e23fe-da74-4a35-8fe3-93bfaaef95f0

name: bidder1101

computeResourceName: rfs.xsmall

templateResourceName: bidderbase

securityResourceName: Web

state: ONLINE

vms:

(empty array)

For further details please consult the c3sdk.

https://github.com/diogogmt/c3-sdk-js

**Puppet Agent (Base template)**

#1 Install puppet

#2 Configure puppet

#3 Check if agent can communicate with the master

#4 Register server base template with resource group

New servers created in the group will use the base template

The hostname for the servers will have the following pattern:

VM{number}{groupname}{uuid}

Based on the server name it is possible to configure the catalogs in the puppet master

**Puppet Master**

#1 Configure puppet

#2 Create autosign rule for a given host/domain

$: vim /etc/puppet/autosign.conf

/^.\*bidder.\*\.cs1cloud\.internal$/

#3 Create catalog for nodes

$: vim /etc/puppet/manifests/sites.pp

node /^.\*bidder.\*\.cs1cloud\.internal$/ {

include motd

include ntp

include cdos

}

In the node definition it is possible to include module that will be pushed to the puppet agents.

All the modules will be compiled into a catalog and that will run on whatever nodes match the regex pattern.

Puppet modules

By default all modules are installed in /etc/puppet/modules

The basic structure of a module is the following:

* /manifests/init.pp -> requires a class to match the module name
* /templates -> ERB templates used for dynamic configuration
* /files -> static files
* /lib -> plugins and resource types

Resources:

Puppet Forge: http://forge.puppetlabs.com/

<http://docs.puppetlabs.com/puppet_core_types_cheatsheet.pdf>

http://docs.puppetlabs.com/puppet/latest/reference/lang\_visual\_index.html