Adding a global config - configurable at different scopes

Existing Global configuration parameter settings:

- Currently CS allows setting global configs which are valid for the entire cloud.
- You need to restart the MS everytime you change the global configs.

Changes with 4.2 release

- Configs can be set at different scopes namely account, zone, cluster, or primary storage
- You do not need to restart the MS after changing these settings (most of the times) - NOTE - you need to use the functions below for this behavior.

New methods to enable configs configurable at different scopes(zone, pod, cluster, pool, account etc.):

As part of 4.2 I came with a new interface methods to retrieve configs at different scopes (account, zone, cluster, or primary storage). If not set they inherit from the global configuration values.

From now on any one introducing a new global config **should use the methods** below in ConfigurationServer interface to get the config value based on the scope.

Change in Interface:

New methods are introduced getConfigValue() and getConfigListByScope()

```
public interface ConfigurationServer {
    public static final String Name = "configuration-server";



    /**
     * get the value of configuration parameter at the scope given for the corresponding resource level
(account, zone, cluster, or primary storage)
     * If either scope or resourceId is not specified then get the global value
     * @param name
     *             configuration parameter name
     * @param scope
     *             Scope at which parameter value is defined (account, zone, cluster, or primary storage)
     * @param resourceId
     *             id of the resource (account, zone, cluster, or primary storage)
     * @return value of the configuration parameter
     */
    public String getConfigValue(String name, String scope, Long resourceId);

    /**
     * get the list of configuration parameters and values at the scope given for the corresponding resource
level (account, zone, cluster, or primary storage)
     * If either scope or resourceId is not specified then get the global value
     * @param scope
     *             Scope at which parameter value is defined (account, zone, cluster, or primary storage)
     * @param resourceId
     *             id of the resource (account, zone, cluster, or primary storage)
     * @return list of the configuration parameters and values
     */
    public List<ConfigurationVO> getConfigListByScope(String scope, Long resourceId);
}
```

Get the value of a parameter at particular scope

@Inject ConfigurationServer _configServer;

1) Get the value of "mem.overprovisioning.factor" defined at cluster (id=3)

overProvisioinigFactor = Float.parseFloat(_configServer.getConfigValue(Config.MemOverprovisioningFactor.key(), Config.ConfigurationParameterScope.cluster.toString(), clusterId));

2) Get the value of "mem.overprovisioning.factor" defined at global level

overProvisioinigFactor = Float.parseFloat(_configServer.getConfigValue(Config.MemOverprovisioningFactor.key(), null, null));

Create parameter at particular scope

We have defined an enum to mention the scope in Config.java

```
public static enum ConfigurationParameterScope {
    global,
    zone,
    cluster,
    storagepool,
    account
}
```

Just add the corresponding scope at the end of the configuration parameter enum value.

Example: **MemOverprovisioningFactor("Advanced", ManagementServer.class, String.class, "mem.overprovisioning.factor", "1", "Used for memory overprovisioning calculation", null, ConfigurationParameterScope.cluster.toString()),**

Update value of configuration parameter:

To update a configuration parameter at particular scope

API name	Existing API Parameters	New optional API Parameters	API Response
updateConfiguration	name: the name of the configuration value: the value of the configuration	zoneid: ID of the zone clusterid: ID of the cluster storageid: ID of the storagepool accountid: ID of the acount	updateconfigurationresponse with scope and updated value

Example: localhost:8096/client/api?command=updateConfiguration&name=mem.overprovisioning.factor&value=2&clusterid=a69d292a-c1e7-4630-8acf-58e955859ab6 (http://

To list configuration parameter at particular scope

API name	Existing API Parameters	New optional API Parameters	API Response
listConfiguration	category: lists configurations by category name: lists configuration by name	zoneid: ID of the zone clusterid: ID of the cluster storageid: ID of the storagepool accountid: ID of the acount	listconfigurationsresponse with scope and value at the corresponding scope of the parameters.

Example: To list the parameter values defined at zone id b7aa7e8c-78b1-49f2-9876-aa25a728276d

http://localhost:8096/client/api?command=listConfigurations&zoneid=b7aa7e8c-78b1-49f2-9876-aa25a728276d

TO DO:

1) Improve the framework to support defining a parameter at multiple scopes (account, zone, cluster, or primary storage)

2) To maintain a local cache of configuration parameter values. All consumers of config parameters should access config values from this cache. Cache should get updated whenever the value in DB is gets updated.