CLASSEC Development

# Developer < Navindra Kumar>

# Story

1. **Generating Authentication Token for HOS:**

Developed code for getting authentication for Helion OpenStack from scratch.

1. **Upgrade getting VM’s list code for HOS:**

Upgraded code for getting list of all virtual machines created on HOS environments.

1. **Updating user credential file for each cloud provider:**

Developed code for updating user configuration file for each cloud environments through API, and also implement permission for it through “jobcof.json”.

1. **Reading user credential file for each cloud provider:**

Developed code for reading user configuration file for each cloud environments through API, and also hiding passwords from being displayed to anyone.

1. **Generating Authentication Token for OpenStack:**

Developed code for getting authentication for OpenStack from scratch initially un-scoped for v3 endpoint. Further upgraded this code for v3/v2.0 endpoint for both scoped and un-scoped authentication Token

1. **Getting VM’s list for OpenStack:**

Developed code for getting list of all virtual machines created on OpenStack environments.

1. **Inserting VM’s details for OpenStack in database:**

Developed code for inserting list of all virtual machines on OpenStack environments into database.

# Tasks

1. Created a configuration file to save data required from user for generating authentication token.
2. Read the configuration file to get authentication endpoint, generate authentication request body and to make authentication token request.
3. Read response body and extract authentication token and endpoint for services like nova, compute metering etc.
4. Updated code to get list of all flavors on HOS environment by replacing hardcoded endpoint.
5. Updated code to get list of all VM’s on HOS environment by replacing hardcoded endpoint.
6. Developed a response structure which contains all required data with proper flavor details of each VM.
7. Developed separate function/code in single file to read request body for each cloud environment.
8. Developed a function code in same file to write the configuration details to the specified file.
9. Developed API for each cloud environment to update configuration with permission check.
10. Developed separate function/code in single file to read configuration details for each cloud environment.
11. Developed API for each cloud environment to get configuration details.
12. Developed code to get un-scoped authentication for new OpenStack Environment for v3 endpoint.
13. Upgraded above code for both scoped and un-scoped authentication token with each v3 and v2.0 endpoints.
14. Developed structure for endpoints returned in request for authentication token.
15. Developed code to get flavors list of OpenStack without hardcoding endpoints and proper error handling.
16. Developed code to get VM’s list of OpenStack without hardcoding endpoints and proper error handling.
17. Developed code for getting list of all flavors and mapped the response with a proper structure.
18. Developed code for getting list of all VM’s and mapped the response with a proper structure, and used the flavors response to put proper details of flavors in VM’s details.
19. Added a function ‘InsertInstances’ to update the OpenStack environments VM’s list in database.

# Environment

Go version go1.7

# Modules impacted

None

# Files added

**HOS**

1. classec\src\gclassec\conf\hosconfiguration.json
2. classec\src\gclassec\goclienthos\authtoken\getAuthToken.go
3. classec\src\gclassec\goclienthos\compute\compute.go
4. classec\src\gclassec\goclienthos\compute\flavors.go

**Configuration update and read**

1. classec\src\gclassec\authmanagment\configurationupdater.go
2. classec\src\gclassec\authmanagment\configurationreader.go

**OpenStack**

1. classec\src\gclassec\openstackgov\authenticationtoken\getauthtoken.go
2. classec\src\gclassec\openstackgov\authenticationtoken\scopedauthtoken.go
3. classec\src\gclassec\openstackgov\authenticationtoken\unscopedauthtoken.go
4. classec\src\gclassec\openstackgov\authenticationtoken\authtoken\_v2.0.go
5. classec\src\gclassec\openstackgov\authenticationtoken\authtoken\_v3.0.go

# Files changed

**OpenStack**

1. classec\src\gclassec\dao\openstackinsert\openstack\_insert.go

# Configuration

**HOS**

1. **Configuration file added** –

* classec\src\gclassec\conf\hosconfiguration.json

1. **Attributes**  – IdentityEndpoint username password, projectName, projectID, container,

region, tenantName, tenantID.

**OpenStack**

1. **Configuration file added** –

* classec\src\gclassec\conf\openstackconfig.json

1. **Attributes**  – identityEndpoint username password, domain, projectName, projectID, container,

region, tenantName, tenantID.

# Dependencies (Libraries)

Basic go libraries are need to be imported in each task.

**HOS**

1. gclassec/goclienthos/authtoken
2. gclassec/structs/hosstruct

**Configuration update and read**

1. gclassec/structs/hosstruct
2. gclassec/structs/openstackInstance
3. gclassec/structs/vmwarestructs
4. gclassec/structs/azurestruct
5. gclassec/structs/awsstructs

**OpenStack**

1. gclassec/structs/openstackInstance
2. gclassec/confmanagement/readopenstackconfig
3. gclassec/openstackgov/ceilometer**"**

# Testcases added

1. Test whether the codes work for different user credentials or not.
2. Test that the compute and flavor request will get endpoints by itself and the proper request is generated for each.
3. Test weather Update user credentials API work for each cloud environment.
4. Test weather after disabling update permission in “jobconf.json” configuration file will not change/update through API’s.
5. Test weather user is able to get configuration details by API, and the sensitive information like password, clientsecret are hidden i.e. not disclosed.
6. Test for new OpenStack environment user will be able to get authentication token depending upon the “IdentityEndpoint” specified in ‘openstackconfig.json’ file. Also he would be able to get scoped or un-scoped authentication token depending upon the function call he made.
7. Test for getting list of VM’s in OpenStack environment with proper details of flavors for each VM in list.

# Code coverage

<Code coverage % of developed code>

# challenges faced

**HOS**

1. Faced problem while generating request body for authentication token request, and also while generating a structure for response body.
2. Faced problem while generating structures for response body of flavor and compute API’s.
3. Faced problem in properly generating response structure for compute response with proper details of flavors.

**Configuration update and read**

1. Faced problem in decoding the encoded request body and saving details in proper json format for updating user credentials.
2. Faced problem in sending proper json response of user configuration details.

**OpenStack**

1. Faced problem in generating request body for scoped and un-scoped authentication token with both v3 and v2.0 endpoints.
2. Faced problem in properly mapping response body for four different authentication response if successfully authenticated and in sending proper error message if failed.
3. Faced problem in sending the response body of as return response of different authentication request, solved by creating a new response structure.
4. Faced problem while generating structures for response body of flavor and compute API’s.
5. Faced problem in properly generating response structure for compute response with proper details of flavors.
6. Faced problem in properly mapping structure with database colums.

# Complexity

<>

# Further comments

This will work only on those machines on which HOS certificate is installed.

# Checklist for deployment

<>