1. **Introduction**

1.1. About

This test specification includes Helmfile executor LCM tests

1.2. Scope of Test Object

Describe the scope of LCM operations

1. **Test Environment**

|  |  |
| --- | --- |
| Hardware | A Kubernetes cluster |
| Middleware | Kubernetes, Helm, Helmfile |
| Test target | eric-lcm-helm-executor |

1. **Basic tests, Cluster connection Info REST API**

3.1 Upload a valid cluster connection info

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.ClusterConnectionInfoLcmTest.uploadValidClusterConnectionInfo()

Description:

Accept, store and delete the cluster connection info

Precondition:

* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create an entity of cluster connection info
2. Verify response status is CREATED
3. DELETE previously created entity

3.2 Upload a valid encrypted cluster connection info

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.ClusterConnectionInfoLcmTest.uploadValidClusterConnectionInfoEncrypted()

Description:

Accept, store and delete the encrypted cluster connection info

Precondition:

* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. Prepare encrypted cluster connection info file
2. POST to create an entity of encrypted cluster connection info
3. Verify response status is CREATED
4. DELETE previously created entity
5. **Basic tests, Workload Instance REST API**

4.1 Reject manual rollback of workload instance after instantiate based on helmfile archive

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.testHelmfileManualRollbackAfterInstantiate()

Description:

Instantiate and rollback workload instance

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Prepare tgz archive from folder contents
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance with helmfile1
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. Verify if all required services are deployed
4. PUT to rollback workload instance without version
5. Verify if all required services are deployed as it was before last action
6. Verify that response is BAD\_REQUETS
7. POST to terminate previously created workload instance
8. GET to verify if operation has status COMPLETED and type TERMINATE
9. DELETE to delete the workload instance entity

Postcondition:

* Remove cluster connection info

4.2 Reject manual rollback of workload instance after reinstantiate based on helmfile archive

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.testHelmfileManualRollbackAfterReinstantiate()

Description:

Instantiate, terminate, reinstantiate and try to rollback workload instance

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Prepare tgz archive from folder contents
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance with helmfile1
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. POST to terminate created workload instance
4. GET to verify if operation has status COMPLETED and type TERMINATE
5. PUT to reinstantiate workload instance
6. GET to verify if operation has status COMPLETED and type REINSTANTIATE
7. PUT to rollback workload instance without version
8. Verify if all required services are deployed as it was before last action
9. Verify that response is BAD\_REQUETS
10. POST to terminate previously created workload instance
11. GET to verify if operation has status COMPLETED and type TERMINATE
12. DELETE to delete the workload instance entity

Postcondition:

* Remove cluster connection info

4.3 Instantiate workload instance based on helmfile archive with complex enabled operation

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.testInstantiateFunctionalEntityWithHelmfile()

Description:

Instantiate workload instance with grouped enable

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Prepare tgz archive from folder contents
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. Verify if all required services are deployed
4. POST to terminate previously created workload instance
5. GET to verify if operation has status COMPLETED and type TERMINATE
6. DELETE to delete the workload instance entity

Postcondition:

* Remove cluster connection info

4.4 Perform different operations (instantiate, update, rollback, terminate) for workload instance based on helmfile

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.shouldSuccessfullyPerformOperationsForHelmfileFirstFlowWithEncryptedValues()

Description:

Process series of upgrades and rollbacks in different combinations and order according to the first flow with encrypted configuration files.

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Create workload instance request body for update
* Create additional parameters for update
* Prepare tgz archive for instantiate (helmfile1) from folder contents
* Prepare tgz archive for upgrade (helmfile2) from folder contents
* Prepare rollback request
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance with helmfile1, encrypted values and clusterConfigInfo file
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. Verify if all required services are deployed
4. Verify if crd is deployed
5. Verify release ordering
6. Verify if docker secret is in place
7. PUT to update workload instance with helmfile2 (without new values) and clusterConfigInfo file
8. GET to verify if operation has status COMPLETED and type UPGRADE
9. Verify if all required services are deployed
10. POST to terminate previously created workload instance with clusterConfigInfo file
11. GET to verify if operation has status COMPLETED and type TERMINATE
12. Verify if services are uninstalled
13. Verify if docker secret is uninstalled
14. PUT to rollback workload instance without version and with clusterConfigInfo file
15. GET to verify if operation has status COMPLETED and type ROLLBACK
16. Verify if all required services are deployed
17. Verify if docker secret is in place
18. PUT to update workload instance with additional parameters, encrypted values and clusterConfigInfo file
19. GET to verify if operation has status COMPLETED and type UPGRADE
20. Verify if all required services are deployed
21. PUT to rollback workload instance without version and with clusterConfigInfo file
22. GET to verify if operation has status COMPLETED and type ROLLBACK
23. Verify if all required services are deployed
24. PUT to rollback workload instance without version and with clusterConfigInfo file
25. GET to verify if operation has status COMPLETED and type ROLLBACK (to show that rollback after rollback is possible)
26. Verify if all required services are deployed
27. GET to get all versions by workload instance
28. Verify if there are 4 versions
29. GET to get version data by specific version and workload instance
30. Verify that fields are not empty
31. PUT to rollback workload instance with specific version and with clusterConfigInfo file
32. GET to verify if operation has status COMPLETED and type ROLLBACK
33. Verify if all required services are deployed
34. POST to terminate previously created workload instance with clusterConfigInfo file
35. GET to verify if operation has status COMPLETED and type TERMINATE
36. Verify if services are uninstalled
37. DELETE to delete the workload instance entity
38. Verify if docker secret is uninstalled

Postcondition:

* Remove cluster connection info

4.5 Perform different operations (instantiate, update, reinstantiate, rollback, terminate) for workload instance based on helmfile

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.shouldSuccessfullyPerformOperationsForHelmfileSecondFlow()

Description:

Process series of upgrades and rollbacks in different combinations and order according to the second flow

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Create workload instance request body for update
* Create additional parameters for update
* Prepare tgz archive for instantiate (helmfile1) from folder contents
* Prepare tgz archive for upgrade (helmfile2) from folder contents
* Prepare rollback request
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance with helmfile1 without clusterConfigInfo
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. Verify if all required services are deployed
4. Verify if crd is deployed
5. PUT to update workload instance with new values without clusterConfigInfo
6. GET to verify if operation has status COMPLETED and type UPGRADE
7. Verify if all required services are deployed
8. POST to terminate previously created workload instance without clusterConfigInfo
9. GET to verify if operation has status COMPLETED and type TERMINATE
10. Verify if services are uninstalled
11. PUT to reinstantiate workload instance without helmfile, values, additionalParams and clusterConfigInfo
12. GET to verify if operation has status COMPLETED and type REINSTANTIATE
13. Verify if all required services are deployed
14. PUT to rollback workload instance without version and clusterConfigInfo
15. GET to verify if operation has status COMPLETED and type ROLLBACK
16. Verify if all required services are deployed
17. PUT to update workload instance with additionalParams without values, helmfile and clusterConfigInfo
18. GET to verify if operation has status COMPLETED and type UPDATE
19. Verify if all required services are deployed
20. GET to get all versions by workload instance
21. GET to get version data by specific version and workload instance
22. Verify that fields are not empty
23. PUT to rollback workload instance with specific version without clusterConfigInfo file
24. GET to verify if operation has status COMPLETED and type ROLLBACK
25. Verify if all required services are deployed
26. POST to terminate previously created workload instance without clusterConfigInfo
27. GET to verify if operation has status COMPLETED and type TERMINATE
28. Verify if services are unistalled
29. PUT to reinstantiate workload instance without helmfile, values, additionalParams and clusterConfigInfo
30. GET to verify if operation has status COMPLETED and type REINSTANTIATE
31. Verify if all required services are deployed
32. GET to get all versions by workload instance
33. GET to get version data by specific version and workload instance
34. Verify that fields are not empty
35. PUT to rollback workload instance with specific version without clusterConfigInfo file
36. GET to verify if operation has status COMPLETED and type ROLLBACK
37. Verify if all required services are deployed
38. POST to terminate previously created workload instance without clusterConfigInfo file
39. GET to verify if operation has status COMPLETED and type TERMINATE
40. Verify if services are unistalled
41. GET to get all versions by workload instance
42. GET to get version data by specific version and workload instance
43. Verify that fields are not empty
44. PUT to rollback workload instance with specific version without clusterConfigInfo file
45. GET to verify if operation has status COMPLETED and type ROLLBACK
46. Verify if all required services are deployed
47. POST to terminate previously created workload instance without clusterConfigInfo file
48. GET to verify if operation has status COMPLETED and type TERMINATE
49. Verify if services are uninstalled
50. DELETE to delete the workload instance entity

Postcondition:

* Remove cluster connection info

4.6 Instantiate workload instance as individual workload function

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.testInstantiateWithHelmfileAsIndividualWorkloadFunction()

Description:

Instantiate two workload instances on the same time

Precondition:

* Upload a valid cluster connection info
* Create workload instance1 request body
* Create workload instance2 request body
* Prepare tgz archive from folder contents
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance1
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. Verify if all required services are deployed
4. Verify if docker secret for the workload instance1 is installed
5. POST to create and instantiate an entity of workload instance2
6. GET to verify if operation has status COMPLETED and type INSTANTIATE
7. Verify if all required services are deployed
8. Verify if docker secret for the workload instance2 is installed
9. POST to terminate previously created workload instance1
10. GET to verify if operation has status COMPLETED and type TERMINATE
11. DELETE to delete the workload instance1 entity
12. POST to terminate previously created workload instance2
13. GET to verify if operation has status COMPLETED and type TERMINATE
14. DELETE to delete the workload instance2 entity

Postcondition:

* Remove cluster connection info

4.7 Perform different operations (instantiate, update, rollback, terminate) for workload instance based on integration chart

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.shouldSuccessfullyPerformOperationsForIntegrationChartFirstFlow()

Description:

Process series of upgrades and rollbacks in different combinations and order based on integration chart according to the first flow

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Create additional parameters for update
* Prepare tgz archive for instantiate (integration\_chart1) from folder contents
* Prepare tgz archive for upgrade (integration\_chart2) from folder contents
* Prepare rollback request
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance with integration\_chart1, values and clusterConfigInfo file
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. Verify if service is deployed
4. Verify if docker secret is in place
5. PUT to update workload instance with integration\_chart2 (without new values) and clusterConfigInfo file
6. GET to verify if operation has status COMPLETED and type UPGRADE
7. Verify if service is deployed
8. Verify that pod is running
9. PUT to update workload instance with new values and clusterConfigInfo file
10. GET to verify if operation has status COMPLETED and type UPGRADE
11. Verify if service is deployed
12. Verify that pod is terminated
13. POST to terminate previously created workload instance with clusterConfigInfo file
14. GET to verify if operation has status COMPLETED and type TERMINATE
15. Verify that pod is terminated
16. PUT to rollback workload instance without version and with clusterConfigInfo file
17. GET to verify if operation has status COMPLETED and type ROLLBACK
18. Verify if service is deployed
19. Verify that pod is terminated
20. PUT to update workload instance with additional parameters and clusterConfigInfo file
21. GET to verify if operation has status COMPLETED and type UPGRADE
22. Verify if service is deployed
23. Verify that pod is running
24. PUT to rollback workload instance without version and with clusterConfigInfo file
25. GET to verify if operation has status COMPLETED and type ROLLBACK
26. Verify if service is deployed
27. Verify that pod is terminated
28. PUT to rollback workload instance without version and with clusterConfigInfo file
29. GET to verify if operation has status COMPLETED and type ROLLBACK (to show that rollback after rollback is possible)
30. Verify if service is deployed
31. Verify that pod is running
32. GET to get all versions by workload instance
33. Verify if there are 4 versions
34. GET to get version data by specific version and workload instance
35. Verify that fields are not empty
36. PUT to rollback workload instance with specific version and with clusterConfigInfo file
37. GET to verify if operation has status COMPLETED and type ROLLBACK
38. Verify if service is deployed
39. Verify that pod is running
40. POST to terminate previously created workload instance with clusterConfigInfo file
41. GET to verify if operation has status COMPLETED and type TERMINATE
42. Verify if services are uninstalled
43. DELETE to delete the workload instance entity

Postcondition:

* Remove cluster connection info

4.8 Perform different operations (instantiate, update, reinstantiate, rollback, terminate) for workload instance based on integration chart

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.shouldSuccessfullyPerformOperationsForIntegrationChartSecondFlow()

Description:

Process series of upgrades and rollbacks in different combinations and order according to the second flow

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Create workload instance request body for update
* Create additional parameters for update
* Prepare tgz archive for instantiate (integration\_chart1) from folder contents
* Prepare tgz archive for upgrade (integration\_chart2) from folder contents
* Prepare rollback request
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance with integration\_chart1 without clusterConfigInfo
2. GET to verify if operation has status COMPLETED and type INSTANTIATE
3. Verify if service is deployed
4. Verify if docker secret is in place
5. PUT to update workload instance with new values without clusterConfigInfo
6. GET to verify if operation has status COMPLETED and type UPGRADE
7. Verify if service is deployed
8. Verify that pod is running
9. POST to terminate previously created workload instance without clusterConfigInfo
10. GET to verify if operation has status COMPLETED and type TERMINATE
11. Verify if service is uninstalled
12. PUT to reinstantiate workload instance without integration chart, values, additionalParams and clusterConfigInfo
13. GET to verify if operation has status COMPLETED and type REINSTANTIATE
14. Verify if service is deployed
15. Verify that pod is running
16. PUT to rollback workload instance without version and clusterConfigInfo
17. GET to verify if operation has status COMPLETED and type ROLLBACK
18. Verify if service is uninstalled
19. Verify that pod is terminated
20. PUT to update workload instance with additionalParams without values, integration chart and clusterConfigInfo
21. GET to verify if operation has status COMPLETED and type UPDATE
22. Verify if service is uninstalled
23. Verify that pod is terminated
24. GET to get all versions by workload instance
25. GET to get version data by specific version and workload instance
26. Verify that fields are not empty
27. PUT to rollback workload instance with specific version without clusterConfigInfo file
28. GET to verify if operation has status COMPLETED and type ROLLBACK
29. Verify if service is uninstalled
30. Verify that pod is terminated
31. POST to terminate previously created workload instance without clusterConfigInfo
32. GET to verify if operation has status COMPLETED and type TERMINATE
33. Verify if services are unistalled
34. PUT to reinstantiate workload instance without integration chart, values, additionalParams and clusterConfigInfo
35. GET to verify if operation has status COMPLETED and type REINSTANTIATE
36. Verify if service is uninstalled
37. Verify that pod is terminated
38. GET to get all versions by workload instance
39. GET to get version data by specific version and workload instance
40. Verify that fields are not empty
41. PUT to rollback workload instance with specific version without clusterConfigInfo file
42. GET to verify if operation has status COMPLETED and type ROLLBACK
43. Verify if service is deployed
44. Verify that pod is running
45. GET to verify if operation has status COMPLETED and type TERMINATE
46. Verify if service is uninstalled
47. GET to get all versions by workload instance
48. GET to get version data by specific version and workload instance
49. Verify that fields are not empty
50. PUT to rollback workload instance with specific version without clusterConfigInfo file
51. GET to verify if operation has status COMPLETED and type ROLLBACK
52. Verify if service is uninstalled
53. Verify that pod is terminated
54. POST to terminate previously created workload instance without clusterConfigInfo file
55. GET to verify if operation has status COMPLETED and type TERMINATE
56. Verify if service is uninstalled
57. DELETE to delete the workload instance entity

Postcondition:

* Remove cluster connection info

4.9 Instantiate CN Workload through the helmfile builder

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.shouldSuccessfullyInstantiateAndUpdateWorkloadInstanceThroughHelmfileBuilder()

Description:

Instantiate and update workload instance through helmfile builder

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance through Helmfile Builder

2. Verify if all required services are deployed

3. PUT to update workload instance with new chart and clusterConnectionInfo

4. Verify if all required services are deployed

5. POST to terminate previously created workload instance

6. DELETE to delete the workload instance entity

Postcondition:

Remove cluster connection info

4.10 Instantiate CN Workload through the helmfile builder with chart orders

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.shouldSuccessfullyInstantiateAndUpdateWorkloadInstanceThroughHelmfileBuilderWithOrder()

Description:

Instantiate and update workload instance through helmfile builder with charts in specified order

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance through Helmfile Builder

2. Verify if all required services are deployed in a correct way

3. POST to terminate previously created workload instance

4. DELETE to delete the workload instance entity

Postcondition:

Remove cluster connection info

4.11 Instantiate workload instance with creating secret

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.HelmSourceLCMTest.testInstantiateFunctionalWithCreateSecretWithHelmfile()

Description:

Instantiate workload instance with creating tls secret

Precondition:

* Upload a valid cluster connection info
* Create workload instance request body with ingress.yaml and non empty tls section
* Upload a certificate and a private key to Helmfile Executor file system
* Helmfile Executor and its dependency services are in ready state

Test Instruction:

1. POST to create and instantiate an entity of workload instance

2. Verify if all required services are deployed in a correct way

3. Verify if tls secret is in place

4. POST to terminate previously created workload instance

5. DELETE to delete the workload instance entity

Postcondition:

Remove cluster connection info

1. **Basic tests, Backup and Restore REST API**

5.1 Process backup and restore operations

Test Method Name:

com.ericsson.oss.mgmt.lcm.acceptance.tests.BackupAndRestoreTest.shouldSuccessfullyProcessBackupAndRestoreOperations()

Description:

Process backup and restore operations

Precondition:

* BRO service is in ready state

Test Instruction:

1. Check the health of the BRO

2. GET to get backup managers

3. Verify backup managers

4. POST to create backup and verify

5. GET to get created backup and verify if status is COMPLETE

6. POST to export backup to sftp server

7. DELETE to delete backup

8. GET to get backup and verify if it’s empty

9. POST to import backup from sftp server

10. GET to get imported backup and verify if status is COMPLETE

11. PUT to restore backup