[1. Distributed Data Infrastructure test cases 2](#_Toc1)

[1.1 iRODS 2](#_Toc2)

[1.1.1 TD\_DDI\_GEN\_001 2](#_Toc3)

[1.1.2 TD\_DDI\_GEN\_002 2](#_Toc4)

[1.1.3 TD\_DDI\_GEN\_003 2](#_Toc5)

[1.1.4 TD\_DDI\_GEN\_004 3](#_Toc6)

[1.2 EUDAT services testcases 3](#_Toc7)

[1.2.1 TD\_B2\_HDL\_001 3](#_Toc8)

[1.2.2 TD\_B2\_SAFE\_001 4](#_Toc9)

[1.3 APIs testcases 4](#_Toc10)

[1.3.1 TD\_DDI\_API\_STG\_001 4](#_Toc11)

[1.3.2 TD\_DDI\_API\_IRD\_001 4](#_Toc12)

[1.3.3 TD\_DDI\_API\_SSH\_001 5](#_Toc13)

[1.3.4 TD\_DDI\_API\_GRM\_001 6](#_Toc14)

[1.4 Other testcases 6](#_Toc15)

[1.4.1 TD\_DDI\_API\_AUTH\_001 6](#_Toc16)

[2. Authentication and Authorization Infrastructure testcases 7](#_Toc17)

[2.1 System monitoring testcases 7](#_Toc18)

[2.1.1 TD\_AAI\_SYS\_001 7](#_Toc19)

[2.1.2 TD\_AAI\_SYS\_002 7](#_Toc20)

[2.1.3 TD\_AAI\_SYS\_003 7](#_Toc21)

[2.1.4 TD\_AAI\_SYS\_004 8](#_Toc22)

[2.1.5 TD\_AAI\_SYS\_005 8](#_Toc23)

[2.1.6 TD\_AAI\_SYS\_006 8](#_Toc24)

[2.1.7 TD\_AAI\_SYS\_007 8](#_Toc25)

[2.2 Security monitoring testcases 9](#_Toc26)

[2.2.1 TD\_AAI\_SEC\_001 9](#_Toc27)

[2.2.2 TD\_AAI\_SEC\_002 9](#_Toc28)

[2.3 Network monitoring testcases 9](#_Toc29)

[2.3.1 TD\_AAI\_NET\_001 9](#_Toc30)

[2.3.2 TD\_AAI\_NET\_002 10](#_Toc31)

[2.3.3 TD\_AAI\_NET\_003 10](#_Toc32)

[2.3.4 TD\_AAI\_NET\_004 10](#_Toc33)

[2.3.5 TD\_AAI\_NET\_005 11](#_Toc34)

[3. Orchestrator monitoring testcases 11](#_Toc35)

[3.1 Functionality monitoring 11](#_Toc36)

[3.1.1 TD\_DAM\_AVL\_001 11](#_Toc37)

[3.1.2 TD\_DAM\_AVL\_002 12](#_Toc38)

[3.1.3 TD\_DAM\_AVL\_003 12](#_Toc39)

[3.2 Networking testcases 12](#_Toc40)

[3.2.1 TD\_DAM\_NET\_001 12](#_Toc41)

# Distributed Data Infrastructure test cases

## 1.1 iRODS

### 1.1.1 TD\_DDI\_GEN\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test iRODS iCAT Server availability** | | | | |
| **Identifier** | | TD\_DDI\_GEN\_001 | | |
| **Purpose** | | Verify that the iRODS zone at a specific center is available | | |
| **Applicability** | | * Target IP should respond to ICMP ping * The iRODS port 1247 should be reachable * Machine should return appropriate response on PING command on the port 1247 * Implemented by blackbox\_exporter and wired to Grafana | | |
| **Pre-conditions** | |  | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Ping iCAT server IP |  |
| 2 | Check | ICMP echo-reply is received |  |
| 3 | Stimulus | Send “PING” to port 1247 |  |
| 4 | Check | XML message containing “<type>RODS\_VERSION</type>” is received |  |
| **Verdict** | The iCAT server is up and running | | |  |

### 1.1.2 TD\_DDI\_GEN\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test iRODS token broker** | | | | |
| **Identifier** | | TD\_DDI\_GEN\_002 | | |
| **Purpose** | | Verify that the iRODS token broker can be used to obtain a valid OpenID token | | |
| **Applicability** | | * OpenID token is provided by the broker using valid LEXIS User id and password * Robot test case | | |
| **Pre-conditions** | | A known valid user name and password is available, LEXIS AAI is up and running | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Call irods broker API with known valid credentials |  |
| 2 | Check | JWT token is received |  |
| 3 | Stimulus | Try to validate the received token |  |
| 4 | Check | Token is valid |  |
| **Verdict** | The iRODS token broker is working | | |  |

### 1.1.3 TD\_DDI\_GEN\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Basic iRODS sanity check** | | | | |
| **Identifier** | | TD\_DDI\_GEN\_003 | | |
| **Purpose** | | iRODS Zone Sanity check without LEXIS AAI | | |
| **Applicability** | | * iRODS Zone is working independently on the AAI * Robot test case | | |
| **Pre-conditions** | | A known valid iRODS user name and password is available | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Try to obtain an iRODS session for supplied credentials |  |
| 2 | Check | A valid iRODSSession object is returned |  |
| 3 | Stimulus | iput a file to the zone |  |
| 4 | Check | List objects in test collection contains the put file |  |
| 5 | Stimulus | Read the file back |  |
| 6 | Check | Contents of the file is correct |  |
| 7 | Stimulus | Delete the test object from the zone |  |
| 8 | Check | List the test collection and verify that the data object got deleted |  |
| **Verdict** | The iRODS zone provides basic functions correctly | | |  |

### 1.1.4 TD\_DDI\_GEN\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: iRODS zone OpenID authentication** | | | | |
| **Identifier** | | TD\_DDI\_GEN\_004 | | |
| **Purpose** | | iRODS Zone Sanity check with LEXIS AAI | | |
| **Applicability** | | * Lexis AAI can be used to access the iRODS zone * Robot test case | | |
| **Pre-conditions** | | A known valid LEXIS user name and password is available, LEXIS AAI is up and runnign | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Try to obtain a token for supplied credentials |  |
| 2 | Check | A valid JWT token is returned |  |
| 3 | Stimulus | iput a file to the zone |  |
| 4 | Check | List objects in test collection contains the put file |  |
| 5 | Stimulus | Read the file back |  |
| 6 | Check | Contents of the file is correct |  |
| 7 | Stimulus | Delete the test object from the zone |  |
| 8 | Check | List the test collection and verify that the data object got deleted |  |
| **Verdict** | The iRODS zone is successfully integrated with LEXIS AAI | | |  |

## 1.2 EUDAT services testcases

### 1.2.1 TD\_B2\_HDL\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test HANDLE system availability** | | | | |
| **Identifier** | | TD\_B2\_HDL\_001 | | |
| **Purpose** | | Verify that the HANDLE system at a specific center is available | | |
| **Applicability** | | * User can acquire a handle | | |
| **Pre-conditions** | | * HANDLE key and certificate are available | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Try to get a handle for lexis-project.eu |  |
| 2 | Check | Verify that the handle is acquired |  |
| 3 | DELETE | Delete the acquired handle |  |
| **Verdict** |  | | |  |

### 1.2.2 TD\_B2\_SAFE\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test B2SAFE status** | | | | |
| **Identifier** | | TD\_B2\_SAFE\_001 | | |
| **Purpose** | | Verify that B2SAFE is up | | |
| **Applicability** | | * Replicate a dataset * Assign a PID | | |
| **Pre-conditions** | | * Keycloak token | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Trigger replication API to replicate a test dataset |  |
| 2 | Check | Verify that the request id returns “Replication completed” with PID assigned |  |
| 3 | DELETE | Delete the replicated dataset and delete the metadata from source dataset |  |
| **Verdict** |  | | |  |

## 1.3 APIs testcases

### 1.3.1 TD\_DDI\_API\_STG\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Staging API availability** | | | | |
| **Identifier** | | TD\_DDI\_API\_STG\_001 | | |
| **Purpose** | | Verify that the Staging API is up and running in a specific center | | |
| **Applicability** | | * Calling the endpoint to fetch available source/target should work * Send a staging request to verify the celery worker is running properly | | |
| **Pre-conditions** | | * Keycloak token must be available and valid | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Send a GET request to the /info subpath |  |
| 2 | Check | Verify that a list with all the targets is returned |  |
| 3 | Stimulus | Send a staging task with source staging area and target iRODS |  |
| 4 | Check | Verify a task ID is returned |  |
| 5 | Stimulus | Query the staging result continuously by calling the check status endpoint |  |
| 6 | Check | Verify the status returned is “Transfer Completed” |  |
| 7 | Clean | Clean up the staged dataset in iRODS |  |
| **Verdict** | A list containing all the possible source/targets are available & a successful transfer of a small dataset | | |  |

### 1.3.2 TD\_DDI\_API\_IRD\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test iRODS API availability** | | | | |
| **Identifier** | | TD\_DDI\_API\_IRD\_001 | | |
| **Purpose** | | Verify that the iRODS API is up and running in a specific center | | |
| **Applicability** | | * Calling the endpoint to fetch available datasets metadata should work * Creation of dataset should work * Deletion of dataset should work | | |
| **Pre-conditions** | | * Keycloak token must be available and valid, and provide access to the testproject project | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Send a POST request to the /dataset/search/metadata subpath with body “{}” |  |
| 2 | Check | Verify that a list with all the available datasets is returned |  |
| 3 | Stimulus | Send a POST request to /dataset subpath with body  {"push\_method": "empty", "access": "user", "project": "testproject"} |  |
| 4 | Check | Verify the result is HTTP status code 201 and  {"status": "201", "internalID": "**<internalID>**”} |  |
| 5 | Stimulus | Send a DELETE request to /dataset subpath with body  {"internalID": "**<internalID>**”, "access": "user", "project": "testproject"} |  |
| 6 | Check | Verify that the result is HTTP status code 201 and  {"errorString": "Request queued on staging api", "stagingAPI": "**<staging api url>**/delete", "request\_id": "**<requestID>**"} |  |
| 7 | Clean | Connect to staging API to verify that the deletion of **<requestID>** is successful |  |
| **Verdict** | A list containing all the available datasets & a successful creation and deletion of an empty dataset | | |  |

### 1.3.3 TD\_DDI\_API\_SSH\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test SSHFS API availability** | | | | |
| **Identifier** | | TD\_DDI\_API\_SSH \_001 | | |
| **Purpose** | | Verify that the SSHFS API is up and running in a specific center | | |
| **Applicability** | | * Calling the endpoint to create an ssh export should work * Calling the endpoint to remove an ssh export should work | | |
| **Pre-conditions** | | * Keycloak token must be available and valid * A valid <path> in the staging zone * A valid ssh keypair * A valid <host> in the SSHFS export network | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Send a POST request to /ssh/sshfsexport with body  {host: “**<host>**”, “pubkey”: “**<pubkey>**”, “path”: **<path>**}” |  |
| 2 | Check1 | Verify the result is HTTP status code 201 with body  {"user": "<user> ", "sshfs": "<user>@<export host>:"} |  |
| 3 | Stimulus | Create a directory <dir> in <host> and run  sshfs <user>@<export host>: <dir> |  |
| 4 | Check | Verify that the sshfs command succeeds and that the data is available at <dir> |  |
| 5 | Clean | umount the <dir> directory |  |
| 6 | Stimulus | Send a DELETE request to /ssh/sshfsexport with body  {“user”: “<user”>, “path”: “<path>”} |  |
| 7 | Check | Verify that the result is HTTP status code 204 and  empty body |  |
| 8 | Clean | Delete the <dir> directory |  |
| **Verdict** | Correct creation and removal of sshfs export, correct access to the staged data while the export is active. | | |  |

### 1.3.4 TD\_DDI\_API\_GRM\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test GRIDMAP API availability** | | | | |
| **Identifier** | | TD\_DDI\_API\_GRM \_001 | | |
| **Purpose** | | Verify that the gridmap API is up and running in a specific center | | |
| **Applicability** | | * Calling the endpoint to add a distinguished name works * Calling the endpoint to remove a user’s direct access to the iRODS backend works | | |
| **Pre-conditions** | | * Keycloak token must be available and valid * A valid Distinguished Name <dn> | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Send a POST request to /gridftp/gridmap with body  {“dn”: “<dn>”} |  |
| 2 | Check | Verify that an HTTP status code of 201 is returned |  |
| 3 | Stimulus | Use irods tools to access your data using <dn> to authentificate yourself |  |
| 4 | Check | Verify that access to your data is granted |  |
| 5 | Stimulus | Send a DELETE request to /dataset subpath with body “{}” |  |
| 6 | Check | Verify that the result is HTTP status code 204 and empty body |  |
| 7 | Stimulus | Use irods tools to access your data using <dn> to authenticate yourself |  |
| 8 | Check | Verify that access to your data is denied |  |
| 9 | Clean | No cleanup needed |  |
| **Verdict** | Correct creation and deletion of iRODS access via distinguished name, correct distinguished name-based authentification while the permissions are active | | |  |

## 1.4 Other testcases

### 1.4.1 TD\_DDI\_API\_AUTH\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test iRODS API response to unauthenticated requests** | | | | |
| **Identifier** | | TD\_DDI\_API\_AUTH \_001 | | |
| **Purpose** | | Verify that the iRODS API responds correctly to unauthenticated requests | | |
| **Applicability** | | * Calling the endpoint to fetch available datasets metadata without proper authorization should fail | | |
| **Pre-conditions** | | * Keycloak token must be available and valid, and provide access to the testproject project | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | Send a POST request to the /dataset/search/metadata subpath with body “{}” but no token |  |
| 2 | Check | Verify that the result is HTTP status code 401 and  {"errorString": "Invalid Authorization"} |  |
| 3 | Clean | No cleanup needed |  |
| **Verdict** | The check confirms that authorization is enabled in the server | | |  |

# 2. Authentication and Authorization Infrastructure testcases

## 2.1 System monitoring testcases

### 2.1.1 TD\_AAI\_SYS\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Database availability internally** | | | | |
| **Identifier** | | TD\_AAI\_SYS\_001 | | |
| **Purpose** | | Verify that the Database process is up and running | | |
| **Applicability** | | * Check mysqld daemon should be up and running * Ensure listening port is 3306 | | |
| **Pre-conditions** | | * Prometheus system Monitoring (internal check) | | |
|  | | | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Check | Verify process is running |  |
| 2 | Check | Verify that port 3306 is up |  |
| **Verdict** |  | | |  |

### 2.1.2 TD\_AAI\_SYS\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Keycloak availability internally** | | | | |
| **Identifier** | | TD\_AAI\_SYS\_002 | | |
| **Purpose** | | Verify that the Keycloak process is up and running | | |
| **Applicability** | | * Check keycloak daemon should be up and running * Ensure listening port is 443 | | |
| **Pre-conditions** | | * Prometheus system Monitoring (internal check) | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Check | Verify process is running |  |
| 2 | Check | Verify that port 443 is up |  |
| **Verdict** |  | | |  |

### 2.1.3 TD\_AAI\_SYS\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Database replication availability internally** | | | | |
| **Identifier** | | TD\_AAI\_SYS\_003 | | |
| **Purpose** | | Verify that the Database replication process is up and running | | |
| **Applicability** | | * Check MySQL replication daemon should be up and running * Ensure listening port is TCP/4567 | | |
| **Pre-conditions** | | * Prometheus system Monitoring (internal check) | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Check | Verify process is running |  |
| 2 | Check | Verify that port TCP/4567 is up |  |
| **Verdict** |  | | |  |

### 2.1.4 TD\_AAI\_SYS\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Database availability from Keycloak** | | | | |
| **Identifier** | | TD\_AAI\_SYS\_004 | | |
| **Purpose** | | Verify that the Database process is accessible from Keycloak cluster | | |
| **Applicability** | | * Check MySQL replication daemon should be up and running * Ensure listening port is TCP/3306 is reachable from Keycloak cluster | | |
| **Pre-conditions** | | * Prometheus system Monitoring (internal check) | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Check | Verify process is running |  |
| 2 | Check | Verify that port TCP/3306 is up |  |
|  | 3 | Check | Verify that port TCP/3306 is accessible from Keycloak cluster nodes |  |
| **Verdict** |  | | |  |

### 2.1.5 TD\_AAI\_SYS\_005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Infinispan discovery availability internally** | | | | |
| **Identifier** | | TD\_AAI\_SYS\_005 | | |
| **Purpose** | | Verify that the Infinispan process up and running | | |
| **Applicability** | | * Check Infinispan daemon should be up and running * Ensure listening port is TCP/7600 | | |
| **Pre-conditions** | | * Prometheus system Monitoring (internal check) | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Check | Verify process is running |  |
| 2 | Check | Verify that port TCP/7600 is up |  |
| **Verdict** |  | | |  |

### 2.1.6 TD\_AAI\_SYS\_006

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Infinispan communication availability internally** | | | | |
| **Identifier** | | TD\_AAI\_SYS\_006 | | |
| **Purpose** | | Verify that the Infinispan process up and running | | |
| **Applicability** | | * Check Infinispan daemon should be up and running * Ensure listening port is TCP/11222 | | |
| **Pre-conditions** | | * Prometheus system Monitoring (internal check) | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Check | Verify process is running |  |
| 2 | Check | Verify that port TCP/11222 is up |  |
| **Verdict** |  | | |  |

### 2.1.7 TD\_AAI\_SYS\_007

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Infinispan availability from Keycloak** | | | | |
| **Identifier** | | TD\_AAI\_SYS\_007 | | |
| **Purpose** | | Verify that the Infinispan process is accessible from Keycloak cluster | | |
| **Applicability** | | * Check Infinispan daemon should be up and running * Ensure listening port is TCP/11222 is reachable from Keycloak cluster | | |
| **Pre-conditions** | | * Prometheus system Monitoring (internal check) | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Check | Verify process is running |  |
| 2 | Check | Verify that port TCP/11222 is up |  |
|  | 3 | Check | Verify that port TCP/11222 is accessible from Keycloak cluster nodes |  |
| **Verdict** |  | | |  |

## 2.2 Security monitoring testcases

### 2.2.1 TD\_AAI\_SEC\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Keycloak Certificate** | | | | |
| **Identifier** | | TD\_AAI\_SEC\_001 | | |
| **Purpose** | | Verify that the Keycloak certificate is valid | | |
| **Applicability** | | * Check Keycloak certificate validity * Raise warning if expiration is lower than 3 months * Raise alert f expiration is lower than 1 month | | |
| **Pre-conditions** | | * Network check from system Monitoring | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | HTTPS Connect port 443 and retrieve certificate |  |
| 2 | Check | Check Certificate validity |  |
| **Verdict** |  | | |  |

### 2.2.2 TD\_AAI\_SEC\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test** | | | | |
| **Identifier** | | TD\_AAI\_SEC\_002 | | |
| **Purpose** | | Verify that the Keycloak certificate is valid | | |
| **Applicability** | | * Check Keycloak certificate validity * Raise warning if expiration is lower than 3 months * Raise alert f expiration is lower than 1 month | | |
| **Pre-conditions** | | * Network check from system Monitoring | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | HTTPS Connect port XXX and retrieve certificate |  |
| 2 | Check | Check Certificate validity |  |
| **Verdict** |  | | |  |

## 2.3 Network monitoring testcases

### 2.3.1 TD\_AAI\_NET\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Database availability externally** | | | | |
| **Identifier** | | TD\_AAI\_NET\_001 | | |
| **Purpose** | | Verify that the Database process is up and running | | |
| **Applicability** | | * Check mysqld daemon should be up and running * MySQL can be reach from monitoring system | | |
| **Pre-conditions** | | * Network check from system Monitoring | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | TCP Connect port 3306 |  |
| 2 | Check | Verify that port 3306 is accessible from network |  |
| 3 | Stimulus | Perform SQL query for “show statistics” |  |
| 4 | Check | Ensure proper response |  |
| **Verdict** |  | | |  |

### 2.3.2 TD\_AAI\_NET\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Keycloak availability externally** | | | | |
| **Identifier** | | TD\_AAI\_NET\_002 | | |
| **Purpose** | | Verify that the Keycloak process is up and running | | |
| **Applicability** | | * Check Keycloak daemon should be up and running * Keycloak can be reach from monitoring system | | |
| **Pre-conditions** | | * Network check from system Monitoring | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | HTTPS Connect port 443 |  |
| 2 | Check | Verify that port 443 is accessible from network |  |
| 3 | Stimulus | Check HTTPS response code |  |
| 4 | Check | Ensure proper response |  |
| **Verdict** |  | | |  |

### 2.3.3 TD\_AAI\_NET\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Database replication availability externally** | | | | |
| **Identifier** | | TD\_AAI\_NET\_003 | | |
| **Purpose** | | Verify that the Database replication process is up and running | | |
| **Applicability** | | * Check MySQL daemon should be up and running * Check MySQL replication daemon can be reach by all nodes of the cluster 4567/tcp 4567/udp 4568/tcp 4444/tcp | | |
| **Pre-conditions** | | * Network check from cluster nodes | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | TCP Connect port 4567, 4568, 4444 |  |
| 2 | Check | Verify that port 4567, 4568, 4444 are accessible from database cluster nodes |  |
| 3 | Check | Ensure proper response |  |
| 4 | Stimulus | UDP Connect port 4567 |  |
| 5 | Check | Verify that port 4567 is accessible from database cluster nodes |  |
| 6 | Check | Ensure proper response |  |
| **Verdict** |  | | |  |

### 2.3.4 TD\_AAI\_NET\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Infinispan availability externally from Database Cluster** | | | | |
| **Identifier** | | TD\_AAI\_NET\_004 | | |
| **Purpose** | | Verify that the Infinispan process is up and running | | |
| **Applicability** | | * Check Infinispan daemon should be up and running * Infinispan can be reach from can be reach by all nodes of the Database cluster | | |
| **Pre-conditions** | | * Network check from system Monitoring | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | TCP Connect port 7600 |  |
| 2 | Check | Verify that port 7600 is accessible from network |  |
| 3 | Check | Ensure proper response |  |
| **Verdict** |  | | |  |

### 2.3.5 TD\_AAI\_NET\_005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test Infinispan availability externally from Keycloak Cluster** | | | | |
| **Identifier** | | TD\_AAI\_NET\_005 | | |
| **Purpose** | | Verify that the Infinispan process is up and running | | |
| **References** | |  | | |
| **Applicability** | | * Check Infinispan daemon should be up and running * Infinispan can be reach from can be reach by all nodes of the Keycloak cluster | | |
| **Pre-conditions** | | * Network check from system Monitoring | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | TCP Connect port 11222 |  |
| 2 | Check | Verify that port 11222 is accessible from network |  |
| 3 | Check | Ensure proper response |  |
| **Verdict** |  | | |  |

# 3. Orchestrator monitoring testcases

## 3.1 Functionality monitoring

### 3.1.1 TD\_DAM\_AVL\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Accessing the VM hosting Dynamic Allocation Module for management of the BL** | | | | |
| **Identifier** | | TD\_DAM\_AVL\_001 | | |
| **Purpose** | | Verify that VM hosting Dynamic Allocation Module is accessible via ssh service | | |
| **References** | |  | | |
| **Applicability** | | * The VM hosting the Dynamic Allocation Module should be running and reachable via VPN | | |
| **Pre-conditions** | | * Enabling the VPN for getting access to the local network where the VM is connected to * Access requires a valid user (i.e., **<USER>**) and secure key (i.e., **<RSA\_KEY>**) to log in | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | ssh -i **<RSA\_KEY>** **<USER>**@<DAM\_IP> whoami | **<USER>** |
| 2 | Check | The virtual machine is reachable |  |
| 3 | Clean | No clean up action is required |  |
| **Verdict** | The test confirms that the DAM hosting VM is up and running | | |  |

### 3.1.2 TD\_DAM\_AVL\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Checking that Dynamic Allocation Module is up and running** | | | | |
| **Identifier** | | TD\_DAM\_AVL\_002 | | |
| **Purpose** | | Verify that the DAM (Flask server) is up and running | | |
| **Applicability** | | * The Compute Cloud should be reachable (the DAM module is hosted on a VM), and the port 9000 should be reachable * Simple HTTP-Rest GET request is performed | | |
| **Pre-conditions** | | Connecting to the DAM hosting VM via ssh | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | curl <DAM\_IP>:9000/alive |  |
| 2 | Check | The DAM module return a json signaling that it is up and running |  |
| 3 | Clean | No clean up action is required |  |
| **Verdict** | The test confirms that the DAM module is up and running | | |  |

### 3.1.3 TD\_DAM\_AVL\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Verifying that the Dynamic Allocation Module is able to access InfluxDB** | | | | |
| **Identifier** | | TD\_DAM\_AVL\_003 | | |
| **Purpose** | | * Verify that the InluxDB instance is up and running * Verify that DAM (Flask server) is able to reach the InfluxDB instance | | |
| **Applicability** | | * The Compute Cloud should be reachable (the DAM module is hosted on a VM), and the port 9000 should be reachable * Terminal-level command to check availability of InfluxDB (port 8086 should be reachable) * Simple HTTP-Rest GET request is performed | | |
| **Pre-conditions** | | Connecting to the DAM hosting VM via ssh | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | influx |  |
| 2 | Check | The local VM InfluxDB client is able to connect to the InfluxDB service on port 8086 |  |
| 3 | Clean | No clean up action is required |  |
|  | 4 | Stimulus | curl "http://<DAM\_IP>:9000/load/netperf/1?src=barbora&dst=salomon" |  |
|  | 5 | Check | The DAM should respond with a valid json |  |
|  | 6 | Clean | No clean up action is required |  |
| **Verdict** | The test confirms that the InfluxDB instance is up and running and that the DAM is able to contact the instance and access the DB(s) | | |  |

## 3.2 Networking testcases

### 3.2.1 TD\_DAM\_NET\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description: Test VM hosting Dynamic Allocation Module** | | | | |
| **Identifier** | | TD\_DAM\_NET\_001 | | |
| **Purpose** | | Verify that VM hosting Dynamic Allocation Module is up and reachable | | |
| **Applicability** | | * The VM hosting the Dynamic Allocation Module should be running and reachable via VPN | | |
| **Pre-conditions** | | * Enabling the VPN for getting access to the local network where the VM (DAM ) is connected to | | |
| **Sequence** | **Step** | **Type** | **Description** | **Result** |
| 1 | Stimulus | telnet <DAM\_IP> 22 |  |
| 2 | Check | The virtual machine is reachable and telnet is able to connect to the ssh service of the VM |  |
| 3 | Clean | No clean up action is required |  |
| **Verdict** | The test confirms that the DAM hosting VM is up and running | | |  |