

Red Hat Confidential

5



redhat.®

JBoss Messaging Release 1.4.x Test Plan

Rev 2
<2011-04-20>

10

<https://docspace.corp.redhat.com/docs/DOC-21118>

15

Miroslav Novák, mnovak@redhat.com
Pavel Slaviček, pslavice@redhat.com

20

Table of Contents

1	Revision history	3
2	Introduction	3
3	Test Priorities	3
4	Test Strategy	3
5	Scope	4
	5.1 Scope – What will be tested?.....	4
	5.2 Scope - What's beyond the scope of testing?.....	4
6	Certified platforms.....	4
7	Test Entrance Criteria, Pass/Fail Criteria, Release Criteria	5
8	Test Deliverables	5
9	Functional test cases.....	6
	9.1 JBoss Messaging test suite.....	6
	9.2 Technology Compatibility Kit (TCK) tests.....	7
10	Integration tests.....	7
	10.1 Correctness of integration check.....	7
	10.2 EAP test-suite	8
	10.3 Transaction crash recovery tests.....	8
11	Fault Injection tests.....	9
	11.1 Failure during acknowledge.....	9
	11.2 Failure during delivering messages via bridge.....	9
	11.3 Failure before commit into database.....	10
12	High-availability tests.....	10
	12.1 Fail-over during server shutdown.....	10
	12.2 Multiple fail-over with no active clients.....	11
	12.3 Network failure.....	12
	12.3.1 Node is disconnected from cluster.....	12
	12.3.2 Node is disconnected from cluster, node is restarted.....	12
	12.3.3 Node is disconnected from cluster, node is restarted but node is still disconnected from cluster.....	13
	12.3.4 Node is disconnected from cluster, connection is restored.....	13
	12.3.5 Node is disconnected from cluster and from database.....	14
	12.3.6 Node is disconnected from database.....	14
	12.3.7 Node is disconnected from database, connection is restored.....	15
	12.3.8 All nodes are disconnected from database.....	16
	12.3.9 All nodes are disconnected from database, connection is restored.....	16
13	Performance and endurance tests.....	17
	13.1 Performance testing.....	17
	13.2 Soak testing.....	17
14	Documentation testing.....	18
	14.1 Technical Review.....	18
	14.2 Examples testing.....	19
15	Test Environment/Configurations	19
16	Responsibilities	19
17	Schedule/Milestones	19
18	Risks and Contingencies	19
19	Approvals	19
20	References	20

1 Revision history

- Rev 1- 2011-04-01 The first draft of the document, document is based on the previous test plan version, <https://docspace.corp.redhat.com/docs/DOC-21118>
- 30 ● Rev 2 – 2011-04-20 Added new test-cases for disconnected node from cluster, updated scenarios according Howard's answers

2 Introduction

The goals of this test plan document are to:

- Define and organize test strategy and test coverage
- 35 ● Communicate the strategy, scope and priorities of the planned tests to all relevant stake-holders for their input and approval
- Scope and priorities of testing
- Test cases definitions and descriptions of the test strategies and scenarios
- Supported platforms

40 3 Test Priorities

The highest priorities for testing are:

- Correctness of JBoss Messaging integration in EAP
- High-Availability
- JBoss Messaging works on all certified platforms/JVMs/databases
- 45 ● Documentation quality
- Performance

4 Test Strategy

Types of Tests / Testing Sequence	<ul style="list-style-type: none"> ● Functional and unit tests – this type of testing will cover basic functions of JBoss Messaging. JBoss Messaging test-suite and some of special test-cases which are important to our customers will be executed. Test suite will be executed across all platforms which are certified for Enterprise application Platform (EAP) 5.1 and for EAP 4.3. https://docspace.corp.redhat.com/docs/DOC-16221. ● Technology Compatibility Kit (TCK) tests – a TCK is a suite of tests, tools and documentation that determines whether or not a product complies with a particular Java™ technology specification. JBoss Messaging provides a fully JMS compliant API. TCK5 JMS 1.1, EJB and EJB3 modules will be executed against JBoss Messaging included in Enterprise application Platform 5.x (EAP) and Enterprise application Platform 4.3. ● Integration tests – main goal of the integration tests is to check correctness of JBoss Messaging integration in Enterprise Application Platform 5.x and Enterprise Application Platform 4.3. Configurations related to integration will be checked manually. EAP test-suite contains basic tests for JMS provider. ● High-availability testing – High availability (HA) is ability for the system to continue functioning after failure of one or more of the servers. A part of high availability is fail-over which is defined as the ability for client connections to migrate from one server to another in event of server failure so client applications can continue to operate. Main goal of this testing is to verify that server is able to provide fail-over for clients. ● Performance testing – main goal of the performance testing is to setup baseline for any future releases. Performance testing is not indeed to produce results for publishing. Performance testing includes tests provided by SPECjms2007 benchmark. ● Soak testing - soak testing involves testing a system with a significant load extended over a significant period of time,
-----------------------------------	---

JBoss Messaging Test Plan

	<p>test will be executed at least for 24h. Configuration value for soak test will be determined from performance testing.</p> <ul style="list-style-type: none">● Documentation testing – main goal of the documentation testing is to check correctness of the distributed documentation and to check if distributed examples are working properly. Documentation should contain description of all supported features, supported configurations.
Test Tools	<ul style="list-style-type: none">● Hudson will be used for tests automation. http://hudson.qa.jboss.com/hudson.● SmartFrog framework will be used. http://wiki.smartfrog.org/wiki/display/sf/SmartFrog+Home, https://docspace.corp.redhat.com/docs/DOC-29753● Performance will be measured by SPECjms2007. SPECjms2007 is the benchmark for evaluating the performance of enterprise message-oriented middle-ware servers based on JMS. http://www.spec.org/jms2007.
Test Processes	<ul style="list-style-type: none">● Test automation, tests will be automated on Hudson, http://hudson.qa.jboss.com/hudson.
Test Reporting	<ul style="list-style-type: none">● Tests reporting, TCMS will be used for test-case tracking, https://tcms.engineering.redhat.com.● Bugs tracking, JIRA will be used for bug tracking, https://jira.jboss.org.

5 Scope

50 5.1 Scope – What will be tested?

Testing will be performed with/for:

- JBoss Messaging test-suite will be executed on all certified platforms/databases/JDBC drivers
- Technology Compatibility Kit (TCK) tests will be executed
- Integration in EAP will be verified
- Automatic client fail-over will be tested
- Basic performance tests will be executed
- Soak test will be executed
- Documentation will be verified

5.2 Scope - What's beyond the scope of testing?

60 Testing will not be performed with/for:

- Project documentation will not be checked
- Platform certification will be provided by JBoss Messaging test-suite and EAP test-suite. All other tests will be executed on RHEL 5/RHEL 6 platforms with Sun JDK
- Results of performance tests will be used only for comparison between releases, they will not be used for publication

6 Certified platforms

JBoss Messaging will be certified on all platforms which are certified for Enterprise Application Platform 5.x and 4.3. Document <https://docspace.corp.redhat.com/docs/DOC-16221> describes supported OSs and JVMs. Document <https://docspace.corp.redhat.com/docs/DOC-16080> describes supported databases and supported JDBC drivers.

OS:

- RHEL 4.x (x86, x86_64)
- RHEL 5.x (x86, x86_64)
- RHEL 6.x (x86, x86_64)
- 75 • Solaris 9 (x86, SPARC32, SPARC64)
- Solaris 10 (x86, SPARC32, SPARC64)
- Windows Server 2003 (x86, x86_64)
- Windows Server 2008 (x86, x86_64)

JVM:

- 80 • Sun JDK 1.6
- IBM JDK 1.6
- Open JDK 1.6

Databases

- IBM DB2 9.7
- 85 • Oracle 10g R2
- Oracle 11g R1
- Oracle 11g R1 RAC
- Oracle 11g R2
- Oracle 11g R2 RAC
- 90 • MySQL 5.0
- MySQL 5.1
- Microsoft SQL Server 2005
- Microsoft SQL Server 2008
- PostgreSQL 8.2
- 95 • PostgreSQL 8.3
- PostgreSQL 8.4
- Sybase ASE 15.0
- Sybase ASE 15.5

7 Test Entrance Criteria, Pass/Fail Criteria, Release Criteria

- 100 Expected results for individual tests will be documented along with the test descriptions in this plan. The overall product entrance criteria and pass/fail criteria are:

JBoss QE Product Release Criteria

<https://docspace.corp.redhat.com/clearspace/docs/DOC-19172>

8 Test Deliverables

- 105 In addition to this test plan and test results documented in TCMS, QE will write “sign off” reviews before the beta and GA release milestones. There will not be a formal beta test for this release, so the beta assessment will be written for the first candidate release (CR) build.

9 Functional test cases

9.1 JBoss Messaging test suite

- 110 Basic functional testing will be covered by JBoss Messaging test-suite. JBoss Messaging project provides huge test-suite which covers provided functions and features. Execution of the JBoss Messaging test-suite will be automated with Hudson on platforms and databases which are certified for Enterprise Application Platform 5.x and 4.3.

JBoss Messaging will be executed on the following reduced OS/JVM/Database matrix:

OS\JVM\Database	Sun JDK 1.6	IBM JDK 1.6	OpenJDK 1.6
RHEL 4 x86	IBM DB2 9.7 PostgreSQL 8.2	Oracle 11g R1 MySQL 5.0	
RHEL 4 x86_64	Oracle 10g R2 MySQL 5.0 Microsoft SQL Server 2008 R2	Oracle 11g R2 RAC IBM DB2 9.7 Sybase ASE 15.5	
RHEL 5 x86	Oracle 11g R1 MySQL 5.1 Microsoft SQL Server 2005	PostgreSQL 8.3 Sybase ASE 15.0	IBM DB2 9.7 Sybase ASE 15.0 PostgreSQL 8.4
RHEL 5 x86_64	Sybase ASE 15.0 PostgreSQL 8.3 PostgreSQL 8.4	Oracle 10g R2 MySQL 5.1	MySQL 5.0 PostgreSQL 8.2 Oracle 10g R2 Sybase ASE 15.5
RHEL 6 x86	Oracle 11g R2 PostgreSQL 8.2	PostgreSQL 8.2 Sybase ASE 15.0	Oracle 11g R1 PostgreSQL 8.3
RHEL 6 x86_64	Oracle 11g R1 RAC Sybase ASE 15.5	PostgreSQL 8.4	Oracle 11g R2 RAC MySQL 5.1
Solaris 9 x86	PostgreSQL 8.3		
Solaris 9 SPARC32	MySQL 5.1		
Solaris 9 SPARC64	Oracle 11g R2		
Solaris 10 x86	MySQL 5.1		
Solaris 10 x86_64	Oracle 10g R2		
Solaris 10 SPARC64	MySQL 5.0		
Windows Server 2003 x86	MySQL 5.1		
Windows Server 2003 x86_64	Microsoft SQL Server 2005		
Windows Server 2008 x86	PostgreSQL 8.3		
Windows Server 2008 x86_64	Microsoft SQL Server 2008 Microsoft SQL Server 2008 R2		

Scenario

- 115 JBoss Messaging project provides build script which runs tests and generates test reports. Testing will be provided according following steps:
1. Download EAP
 2. Checkout of corresponding JBoss Messaging version from project SVN repository
 3. Build JBoss Messaging from sources
- 120
4. Replace all compiled and packaged jars from previous step with corresponding jars from EAP distribution
 5. Run test-suite

Pass criteria

There are no failed test.

9.2 Technology Compatibility Kit (TCK) tests

- 125 A TCK is a suite of tests, tools and documentation that determines whether or not a product complies with a particular Java™ technology specification. JBoss Messaging provides a fully JMS compliant API. TCK JMS module will be executed against Enterprise application Platform 5.x and 4.3.
- TCK EJB module will be executed against Enterprise application Platform 5.x and 4.3.
- TCK EJB 3.0 module will be executed against Enterprise application Platform 5.x.

130 **Scenario**

1. Download EAP
2. Setup EAP and TCK configuration
3. Start test

Pass criteria

- 135 There are no failed tests.

10 Integration tests

10.1 Correctness of integration check

Main goal of this test is to verify correctness of JBoss Messaging integration in EAP. Test will be done manually.

Scenario

140 **Client libraries**

EAP client directory contains all jars required by client.

Deployers

JBoss Messaging deployers are available and configured.

145

JCA adapter

JBoss Messaging resource adapter is named jms-ra.rar. Adapter is deployed in deploy directory of all configuration profiles where is JMS supported.

XA recovery configuration

150

JBoss Messaging's XA Recovery is configured in configuration file located in conf/jbossts-properties.xml of all EAP's configuration profiles where is JMS supported.

AS Administration and JMX Console

It is possible to create and configure JBoss Messaging resources via the Administration and JMX Console.

155

Pass criteria

There is no problem in checked items.

10.2 EAP test-suite

EAP test-suite will be part of integration tests. Test-suite contains several tests for JMS component and takes care of whole product testing.

160

Scenario

Execute full test-suite on all supported platforms and JVM's as part of standard EAP test cycle.

Pass criteria

There are no failed tests.

10.3 Transaction crash recovery tests

165

Main goal of this test is to verify that JBoss Messaging supports XA recovery. XA recovery deals with system or application failures to ensure that of a transaction are applied consistently to all resources affected by the transaction, even if any of the application processes or the machine hosting them crash or lose network connectivity.

Part of this test-case is covered in JTA testing.

Scenario

170

Test initiates a transaction on server by invoking an SLSBs method, enlists into that transaction a special XA resource that is configured to perform a failure during the transaction life-cycle and sends message. When the fault XA resource crashes the server, the test framework recognizes it, restarts the server and checks whether the crashed transaction has been fully recovered. Server will be restarted in the same configuration. This scenario is tested during JTA testing.

The second scenario is be applied on MDB. MDB consumes messages from input queue and re-send messages to output queue. Server will be killed during re-sending messages to output queue.

175

Pass criteria

There are no failed tests, there are no missing messages and no duplicated messages.

11 Fault Injection tests

180 Main aim of this scenario is to test if JBM is able to work correctly even if server is killed at critical point of time. Critical points will be determined by analysis of the implementation, according already reported issues and from discussion with dev team.

JBoss Byteman will be used as fault injection tool. Byteman allows to kill server at defined point of implementation or it allows to create exception.

11.1 Failure during acknowledge

185 Main goal of this test is to simulate server fail when server saves client's acknowledge. Client should receive an exception.

Scenario

1. Start EAP server
2. Send messages into queue with stand-alone client
- 190 3. Install Byteman script, script counts data and if counter reaches specified value server is killed via `jvmskill()`. Counter is increased by each acknowledge.
4. Try to consume all messages with stand-alone client from queue with client acknowledge mode, Byteman kills server
5. Check records in database
6. Restart killed server
- 195 7. Try to consume all messages with stand-alone client from output queue with client acknowledge mode, Byteman kills server

Pass criteria

All messages are delivered, no duplicated messages, there are no orphan messages in database.

11.2 Failure during delivering messages via bridge

200 Main goal of this test is to simulate server fail when server reads message from input queue and tries to send message to output queue.

Scenario

1. Start EAP servers, server A and server B
2. Send messages into input queue with stand-alone client
- 205 3. Install Byteman script, script counts data and if counter reaches specified value server is killed via `jvmskill()`
4. Deploy bridge definition, Byteman kills server
5. Check records in database
6. Restart killed server
7. Try to consume all messages with stand-alone client from output queue

210 **Pass criteria**

All messages are delivered, no duplicated messages, there are no orphan messages in database.

11.3 Failure before commit into database

Main goal of this test is to simulate server fail just before information about new message is committed into database. Server is killed at this specific point of implementation.

215 **Scenario**

1. Start EAP server
2. Install Byteman script, script counts data and if counter reaches specified value server is killed via jvmskill()
3. Send messages to server with stand-alone client, Byteman kills server
4. Check records in database
- 220 5. Restart killed server
6. Try to consume all messages with stand-alone client

Pass criteria

All messages are delivered, no duplicated messages, there are no orphan messages in database.

12 High-availability tests

225 **12.1 Fail-over during server shutdown**

Main aim of the fail-over testing during server shutdown is to check if JBoss Messaging is able to provide fail-over for different types of server shutdown, different types of configuration etc.

Test will be executed with different types of configurations:

Types of server shutdown:

- 230
- Clean server shutdown, normal shutdown process
 - Kill command, server process will be killed by Linux kill command

Types of session:

- Not-transacted session, messages are send via not-transacted session
- Transacted session, messages are send via transacted session, several messages are send in transaction

235 Types of acknowledge modes:

- Auto-acknowledge, client-acknowledge

Server will be configured with “do fail-over on clean shutdown”.

Tested configuration matrix:

Shutdown mode\Session type	Not-transacted session	Transacted session
Kill server	Auto-acknowledge	Group of 100 messages

JBoss Messaging Test Plan

	Client-acknowledge	
Clean shutdown	Auto-acknowledge Client-acknowledge	Group of 100 messages

Scenario

240 Test will be executed on cluster with three nodes. MySQL database will be used as persistent store for JBoss messaging.

1. Setup EAP cluster
2. Start clients (senders, receivers), clients are active during whole test
3. Stop/kill servers from cluster, at least one server must stay alive
4. Restart killed server

- 245
5. Repeat 3. for several times
 6. Stop producers and let's give time to receivers to receive all messages
 7. Stop cluster
 8. Verify results

Pass criteria

250 All messages were delivered, no duplicated messages, no orphan messages in database.

12.2 Multiple fail-over with no active clients

Main aim of the fail-over testing with not active clients is to check if JBoss Messaging is able to provide fail-over and all messages stored in database are correctly re-assigned to another node of the cluster.

Scenario

255 Test will be executed on cluster with two nodes. MySQL database will be used as persistent store for JBoss messaging.

1. Prepare and start cluster of 2 nodes, server A and server B
2. Send 1 000 messages to each node of the cluster
3. Destroy connection between A and B
4. Wait until nodes detect lost connection

- 260
5. Disconnect A from DB and wait for fail-over to B
 6. Disconnect B from DB and wait for B to be shut downed
 7. Reconnect A to DB and wait for node A to start again
 8. Read messages from A

Pass criteria

265 All messages were delivered, no duplicated messages, no orphan messages in database.

12.3 Network failure

Main aim of the following test-cases is to prove correct behavior of JBoss Messaging when is server disconnected from database or when is server disconnected from cluster because of network failure.

270 If test-case does not contain specific description of the test configuration, test is executed on cluster of the four nodes (A, B, C, D) and MySQL database is used as shared persistent storage. Each test uses two clustered queues deployed on each node of the cluster. Queue1 is used for checking of correctness of fail-over mechanism and it is not used by active clients. Queue2 is used by active clients during tests, it is used for checking if all messages were delivered successfully without duplicated messages.

275 Network failure between server and database server is simulated with controllable proxy server which stays between server node and database node. This proxy allows to disconnect communication between two servers. Network failure between nodes in cluster is simulated with special JGroup protocol implementation. Protocol allows to ignore communication from selected nodes of the cluster. List of the ignored nodes of the cluster can be controlled via MBean which allows us to dynamically change this list during test. If node is blocked by this mechanism it does not receive information about cluster from another nodes and node starts to think that it was disconnected from cluster.

280 12.3.1 Node is disconnected from cluster

Node is disconnected from cluster but it keeps its connection to the database server.

Scenario

1. Prepare and start cluster of servers
2. Send 1 000 messages into queue1 on each node
- 285 3. Start active clients
4. Disconnect server A from cluster
5. Wait for some time e.g. 10 mins
6. Stop active clients
7. Stop cluster

290 **Pass criteria**

Fail-over for A node should not occur. No orphan messages, no duplicated messages and all messages should be delivered.

12.3.2 Node is disconnected from cluster, node is restarted

Node is disconnected from cluster but it keeps its connection to the database server, administrator restarts disconnected node.

295 **Scenario**

1. Prepare and start cluster of servers
2. Send 1 000 messages into queue1 on each node
3. Start active clients
4. Disconnect server A from cluster

- 300
5. Wait for some time e.g. 10 mins
 6. Start active client which is connected to A server
 7. Restart disconnected server A, server re-connects with cluster (node will be restarted with active client from step 6, reading/sending messages)
 8. Stop active clients
- 305
9. Stop cluster

Pass criteria

Fail-over for A node should not occur. No orphan messages, no duplicated messages and all messages should be delivered. Active client from step 6 are re-connected to another live node of the cluster.

310 **12.3.3 Node is disconnected from cluster, node is restarted but node is still disconnected from cluster**

Node is disconnected from cluster but it keeps its connection to the database server, administrator restarts disconnected node. But after restart client is not still connected to the cluster (network failure etc.)

Scenario

1. Prepare and start cluster of servers
- 315 2. Send 1 000 messages into queue1 on each node
3. Start active clients
4. Disconnect server A from cluster
5. Wait for some time e.g. 10 mins
6. Start active client which is connected to A server
- 320 7. Restart disconnected server A, server does not re-connects with cluster (node will be restarted with active client from step 6, reading/sending messages), there should be at least 5 min before server shutdown and start.
8. Start new active client which is connected to A server
9. Stop active clients
10. Stop cluster

325 **Pass criteria**

Fail-over for A node should occur but restarted node A should not provide any fail-over for another node of the cluster. No orphan messages, no duplicated messages and all messages should be delivered. Active client from step 6 are re-connected to another live node of the cluster. New active client should be able to read/write all messages via node A from another nodes of the cluster.

330 **12.3.4 Node is disconnected from cluster, connection is restored**

Node is disconnected from cluster but it keeps its connection to the database server. Connection is restored after some time.

Scenario

1. Prepare and start cluster of servers
2. Send 1 000 messages into queue1 on each node
- 335 3. Start active clients
4. Disconnect server A from cluster
5. Wait for some time e.g. 10 mins
6. Restore connection for server A
7. Wait for some time e.g. 10 mins
- 340 8. Stop active clients
9. Stop cluster

Pass criteria

Fail-over for A node should not occur. No orphan messages, no duplicated messages and all messages should be delivered. Client should be able to read all messages from clustered queue when is connection restored.

345 **12.3.5 Node is disconnected from cluster and from database**

Node is disconnected from database and from the cluster. Node detects this situation and JBoss Messaging is stopped automatically on this node.

Scenario

1. Prepare and start cluster of servers
- 350 2. Send 1 000 messages into queue1 on each node
3. Start active clients
4. Disconnect server A from database and from another nodes of the cluster
5. Wait for some time e.g. 5 mins
6. Try to connect new clients via HA-JNDI
- 355 7. Stop clients-producers
8. Stop clients-consumers
9. Stop cluster

Pass criteria

360 All messages were delivered, no duplicated messages, no orphan messages in database. JBoss Messaging is stopped on the A node automatically and client is re-connected to another node of the cluster. If new client tries to connect to node A and uses ClusteredConnectionFactory, client is re-connected to another node of the cluster.

12.3.6 Node is disconnected from database

Node is disconnected from database server but it keeps its connection with another nodes of the cluster. Node will be configured with re-try connection -1.

365 **Scenario**

1. Prepare and start cluster of servers
2. Send 1 000 messages into queue1 on each node
3. Start active clients
4. Disconnect server A from database server
- 370 5. Wait for 5 minutes
6. Stop active clients
7. Stop cluster

Pass criteria

TODO

375 Q: What is expected behavior here? Will be client re-connected to another live node? Disconnected node could not provide required functionality for client.

Q: What will happen with disconnected node?

Q: What is the expected behavior when new client is connected to the disconnected node?

380 Q: Current implementation does not provide fail-over for this disconnected node. Is it expected behavior? We think that fail-over should occur here and clients connected to another nodes of the cluster should be able to read all messages from clustered queue.

12.3.7 Node is disconnected from database, connection is restored

Node is disconnected from database server but it keeps its connection with another nodes of the cluster, after some time is connection to the database restored.

385 **Scenario**

1. Prepare and start cluster of servers
2. Send 1 000 messages into queue1 on each node
3. Start active clients
4. Disconnect server A from database server
- 390 5. Wait for 10 minutes
6. Restore database connection for server A
7. Stop active clients
8. Stop cluster

Pass criteria

395 **TODO**

Q: Same questions as we have for 12.3.4.

Q: If max connection re-try expires and DB connection is restored, node is not reconnected to DB, which is expected

behavior. But it seems that this situation corrupts whole cluster, is it expected behavior?

12.3.8 All nodes are disconnected from database

400 All nodes of the cluster are disconnected from the database server.

Scenario

1. Prepare and start cluster of servers
2. Send 1 000 messages into queue1 on each node
3. Start active clients
- 405 4. Disconnect servers from database server
5. Wait for some time e.g. 5 mins
6. Stop cluster

Pass criteria

Active client receives exception when is database disconnected. Will be investigated.

410 12.3.9 All nodes are disconnected from database, connection is restored

All nodes of the cluster are disconnected from the database server, connection is after some time restored.

Scenario

1. Prepare and start cluster of servers
2. Send 1 000 messages into queue1 on each node
- 415 3. Start active clients
4. Disconnect servers from database server
5. Wait for some time e.g. 5 mins
6. Restore connection to database server
7. Start new active clients
- 420 8. Stop cluster

Pass criteria

Active client receives exception when is database disconnected. Client should be able to read messages when is connection restored. Will be investigated.

13 Performance and endurance tests

425 13.1 Performance testing

Main goal of this test is set performance baseline for future releases. This tests is not intended to prepare results for publication but is intended for checking if there are any regressions in performance between EAP releases.

Test will be executed on EAP's production profile and default hardware configuration will be used. We will not tune server, JVM or hardware configuration to reach better performance.

430 Performance will be measured by SPECjms2007 benchmark test.

Scenario

Test will be automated on Hudson and it will be implemented with SmartFrog framework. Developed SmartFrog components will try to find max value for SPECjms2007 benchmark test and will be responsible for:

- preparation of the test environment on different servers
- 435 • compilation and configuration of SPECjms2007
- starting and stopping EAP server
- starting and stopping SPECjms2007 controller and satellites
- collecting results and results evaluating

Test procedure:

- 440 1. Prepare EAP server, use MySQL database for JBM as persistent storage
2. Install JBoss Messaging JMS provider classes into SPECJjms2007 and compile distribution
3. Cleanup test environment
4. Configure SPECjms2007 with desired value
5. Start EAP server
- 445 6. Start SPECjms2007 jms setup phase
7. Start SPECjms2007 satellites
8. Start SPECjms2007 agent and execute test
9. Stop EAP server
10. Collect and evaluate results
- 450 11. If test should continue, calculate new configuration value and go to 3.
12. End

Pass criteria

There is no significant (>5%) performance degradation in comparison with previous release. Small degradation can be caused by OS, network etc.

455 13.2 Soak testing

The purpose of this test is to verify that there are no performance or functionality regressions in server while running over

extended period of time under significant load. SPECjms2007 will be used for soak testing. SPECjms2007 will be configured to provide significant load to the server. Utilization of servers resources will be measured during the test, CPU load and JVM memory.

460 **Scenario**

Test will be executed with one EAP server started in production profile. Test will be automated on Hudson and it will be implemented with SmarfFrog framework. SmarfFrog components which were developed for performance testing will be reused for this test.

465 Test will be executed for 1 hour and for 72 hours and it will be executed on same hardware as it was used for performance testing. 1 hour test will be used for smoke check, 72 hours soak will be executed only for one time during whole release testing. SPECjms2007 will be configured to produce significant load which was measured during performance testing. Configured load will be same during the whole test.

Pass criteria

470 Performance stabilizes without increasing CPU and JVM memory utilization and remains constant. There is no evident memory leak and no signs of potential problems.

14 Documentation testing

Documentation testing is divided into two phases, Technical Review of documentation and Examples testing. Main aim of the Technical Review is to check shipped documentation, main aim of Examples testing is check if all shipped examples works properly.

475 **14.1 Technical Review**

The purpose of Technical Review is to identify any flaws with the content of the document, not with its presentation or with grammatical correctness.

Scenario

Things that Technical Reviewers should be looking for include:

- 480 • content that no longer applies to the current version
- content that does not apply to the product, such as configuration or installation/deployment instructions
- references to items, such as code examples, that do not ship with the product
- out-of-date screen shots
- incomplete procedures (e.g. missing or assumed steps)
- 485 • references to old on-line content
- documentation for no supported features

Pass criteria

- Documentation is available in requested formats and requested formatting which is usual for EAP's documentation
- Documentation does not include any items which have been listed in Scenario

490 **14.2 Examples testing**

JBoss Messaging examples are included in documentation. The main goal of examples testing is to check if included examples are working properly with Enterprise Application Platform.

Scenario

All included examples will be executed according attached instructions.

495 **Pass criteria**

All examples works properly.

15 Test Environment/Configurations

Tests will be executed in the following Labs:

- JBoss Messaging test-suites will be executed on required platforms, all required platforms are available as Hudson's slave servers
- Performance, HA and SOAK tests will be executed in Messaging Lab

16 Responsibilities

TODO

17 Schedule/Milestones

505 See: <https://engineering.redhat.com/pp/product/>

QA Reference: <https://docspace.corp.redhat.com/docs/DOC-19049>

18 Risks and Contingencies

- Tests development
 - Test plan is based on the old version of the test plan but it contains several new test-cases which are not yet developed
- Testing
 - Test execution will consume lot of time and resources
 - Test plan contains new test-cases where we can find new blockers for release
 - Performance testing will have to be executed on the same hardware to get comparable results
 - There are no dedicated servers for JBoss Messaging testing, time of servers will be shared for testing of other EAP components during the test cycle

19 Approvals

Reviewer	Representing
-----------------	---------------------

JBoss Messaging Test Plan

Bruno Georges	TDM
Rich Sharples	Product Manager
Kimberly Palko	Product Manager
Anne-Louise Tangring	Program Manager
Mike Harvey	QE Manager
Rajesh Rajasekaran	QE Lead
Fernando Nasser	Productization Lead
Laura Bailey	Documentation Lead
Jimmy Wilson	Support Lead
Clebert Suconic	Development Lead
Mark Little	JBoss CTO
Andy Miller	VP Eng.
Jim Moran	Support Management

20 References

- EAP5 Testing - OS/JVM Matrix, <https://docspace.corp.redhat.com/docs/DOC-16221>
- JBoss Messaging documentation

520