

Testing High-Availability Deployments

About Terracotta Documentation

This documentation is about Terracotta DSO, an advanced distributed-computing technology aimed at meeting special clustering requirements.

Terracotta products without the overhead and complexity of DSO meet the needs of almost all use cases and clustering requirements. To learn how to migrate from Terracotta DSO to standard Terracotta products, see [Migrating From Terracotta DSO](#). To find documentation on non-DSO (standard) Terracotta products, see [Terracotta Documentation](#). Terracotta release information, such as release notes and platform compatibility, is found in [Product Information](#).

- **Introduction**
- [How DSO Clustering Works](#)
- [Platform Concepts](#)
- [Hello Clustered World](#)
- **Setup and Configuration**
- [Planning for a Clustered App](#)
- [Configuring Terracotta DSO](#)
- [Configuration Reference](#)
- [Installation](#)
- **APIs**
- [Using Annotations](#)
- [Cluster Events](#)
- [Data Locality Methods](#)
- [Distributed Cache](#)
- [Clustered Async Data Processing](#)
- **Tool Guides**
- [Developer Console](#)
- [Operations Center](#)
- [tim-get \(TIM Management Tool\)](#)
- [Platform Statistics Recorder](#)
- [Eclipse Plugin](#)
- [Sessions Configurator](#)
- [Clustering Spring Webapp with Sessions Configurator](#)
- [Maven](#)
- [JMX](#)
- **Testing, Tuning, and Deployment**
- [Top 5 Tuning Tips](#)
- [Testing a Clustered App](#)
- [Tuning a Clustered App](#)
- [Deployment Guide](#)
- [Operations Guide](#)
- **FAQs and Troubleshooting**
- [General FAQ](#)
- [DSO Technical FAQ](#)
- [Troubleshooting Guide](#)
- [Gotchas](#)
- [Non-portable Classes](#)
- **Reference**
- [Migrating From DSO](#)
- [Concept and Architecture Guide](#)
- [Examinator Reference Application](#)
- [Clustered Data Structures Guide](#)
- [Integrating Terracotta DSO](#)
- [Clustering Spring Framework](#)
- [Integration Modules Manual](#)
- [AspectWerkz Pattern Language](#)
- [Glossary](#)

Release: 3.6

Publish Date: November, 2011 [Documentation Archive](#) »

Testing High-Availability Deployments

- High Availability Network Architecture And Testing
- Deployment Configuration: Simple (no network redundancy)
 - Description
 - Additional configuration
 - Test Plan - Network Failures Non-Redundant Network
 - Test Plan - Network Tests Non-redundant Network
- Deployment Configuration: Fully Redundant
 - Description
 - Additional configuration
 - Test Plan - Network Failures Redundant Network
 - Test Plan - Network Testing Redundant Network
- Terracotta Cluster Tests
 - Test Plan - Active L2 System Loss Tests - verify Passive Takeover
 - Test Plan - Passive L2 System Loss Tests
 - Test Plan - Failover/Failback Tests
 - Test Plan - Loss of Switch Tests
 - Test Plan - Loss of Network Connectivity
 - Test Plan - Terracotta Cluster Failure
 - Client Failure Tests

High Availability Network Architecture And Testing

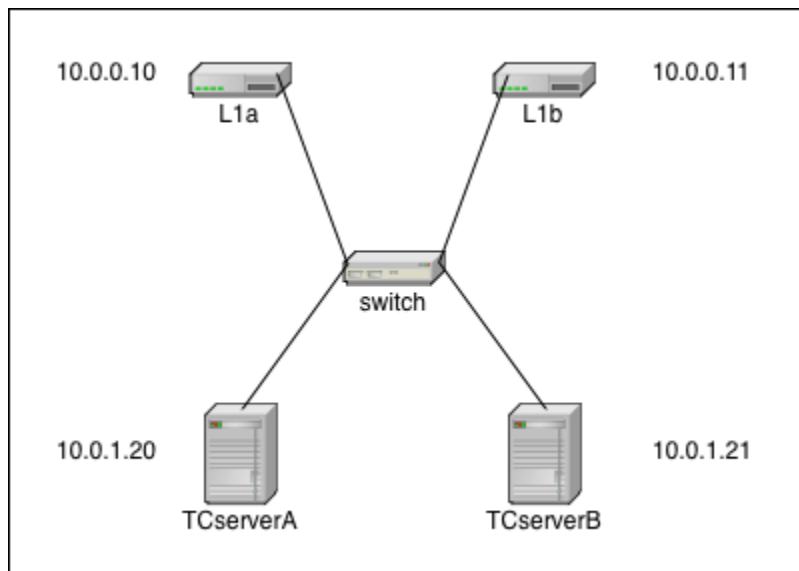
To take advantage of the Terracotta active-passive server configuration, certain network configurations are necessary to prevent split-brain scenarios and ensure that Terracotta clients (L1s) and server instances (L2s) behave in a deterministic manner after a failure occurs. This is regardless of the nature of the failure, whether network, machine, or other type.



If you've turned off disk caching to prevent loss of data in case of a power outage to all Terracotta server instances in the cluster, performance may suffer substantial degradation. See the [this troubleshooting issue](#) for more information.

This document outlines two possible network configurations that are known to work with Terracotta failover. While it is possible for other network configurations to work reliably, the configurations listed in this document have been well tested and are fully supported.

Deployment Configuration: Simple (no network redundancy)



Description

This is the simplest network configuration. There is no network redundancy so when any failure occurs, there is a good chance that all or part of the cluster will stop functioning. All failover activity is up to the Terracotta software.

In this diagram, the IP addresses are merely examples to demonstrate that the L1s (L1a & L1b) and L2s (TCserverA & TCserverB) can live on different subnets. The actual addressing scheme is specific to your environment. The single switch is a single point of failure.

Additional configuration

There is no additional network or operating-system configuration necessary in this configuration. Each machine needs a proper network configuration (IP address, subnet mask, gateway, DNS, NTP, hostname) and must be plugged into the network.

Test Plan - Network Failures Non-Redundant Network

To determine that your configuration is correct, use the following tests to confirm all failure scenarios behave as expected.

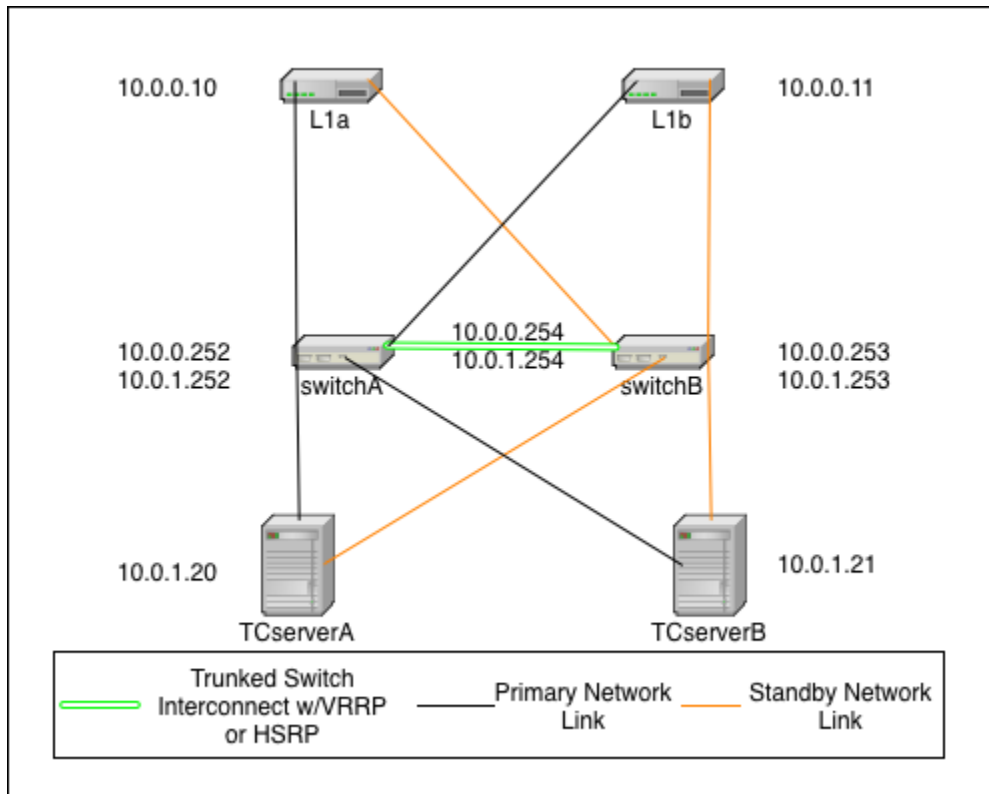
TestID	Failure	Expected Outcome
FS1	Loss of L1a (link or system)	Cluster continues as normal using only L1b
FS2	Loss of L1b (link or system)	Cluster continues as normal using only L1a
FS3	Loss of L1a & L1b	Non-functioning cluster
FS4	Loss of Switch	Non-functioning cluster
FS5	Loss of Active L2 (link or system)	Passive L2 becomes new Active L2, L1s fail over to new Active L2
FS6	Loss of Passive L2	Cluster continues as normal without TC redundancy
FS7	Loss of TCservers A & B	Non-functioning cluster

Test Plan - Network Tests Non-redundant Network

After the network has been configured, you can test your configuration with simple ping tests.

TestID	Host	Action	Expected Outcome
NT1	all	ping every other host	successful ping
NT2	all	pull network cable during continuous ping	ping failure until link restored
NT3	switch	reload	all pings cease until reload complete and links restored

Deployment Configuration: Fully Redundant



Description

This is the fully redundant network configuration. It relies on the failover capabilities of Terracotta, the switches, and the operating system. In this scenario it is even possible to sustain certain double failures and still maintain a fully functioning cluster.

In this diagram, the IP addressing scheme is merely to demonstrate that the L1s (L1a & L1b) can be on a different subnet than the L2s (TCserverA & TCserverB). The actual addressing scheme will be specific to your environment. If you choose to implement with a single subnet, then there will be no need for VRRP/HSRP but you will still need to configure a single VLAN (can be VLAN 1) for all TC cluster machines.

In this diagram, there are two switches that are connected with trunked links for redundancy and which implement Virtual Router Redundancy Protocol (VRRP) or HSRP to provide redundant network paths to the cluster servers in the event of a switch failure. Additionally, all servers are configured with both a primary and secondary network link which is controlled by the operating system. In the event of a NIC or link failure on any single link, the operating system should fail over to the backup link without disturbing (e.g. restarting) the Java processes (L1 or L2) on the systems.

The Terracotta fail over is identical to that in the simple case above, however both NIC cards on a single host would need to fail in this scenario before the TC software initiates any fail over of its own.

Additional configuration

- Switch - Switches need to implement VRRP or HSRP to provide redundant gateways for each subnet. Switches also need to have a trunked connection of two or more lines in order to prevent any single link failure from splitting the virtual router in two.
- Operating System - Hosts need to be configured with bonded network interfaces connected to the two different switches. For Linux, choose mode 1. More information about Linux channel bonding can be found in the [RedHat Linux Reference Guide](#). Pay special attention to the amount of time it takes for your VRRP or HSRP implementation to reconverge after a recovery. You don't want your NICs to change to a switch that is not ready to pass traffic. This should be tunable in your bonding configuration.

Test Plan - Network Failures Redundant Network

The following tests continue the tests listed in Network Failures (Pt. 1). Use these tests to confirm that your network is configured properly.

TestID	Failure	Expected Outcome
FS8	Loss of any primary network link	Failover to standby link
FS9	Loss of all primary links	All nodes fail to their secondary link
FS10	Loss of any switch	Remaining switch assumes VRRP address and switches fail over NICs if necessary
FS11	Loss of any L1 (both links or system)	Cluster continues as normal using only other L1
FS12	Loss of Active L2	Passive L2 becomes the new Active L2, All L1s fail over to the new Active L2
FS13	Loss of Passive L2	Cluster continues as normal without TC redundancy
FS14	Loss of both switches	non-functioning cluster
FS15	Loss of single link in switch trunk	Cluster continues as normal without trunk redundancy
FS16	Loss of both trunk links	possible non-functioning cluster depending on VRRP or HSRP implementation
FS17	Loss of both L1s	non-functioning cluster
FS18	Loss of both L2s	non-functioning cluster

Test Plan - Network Testing Redundant Network

After the network has been configured, you can test your configuration with simple ping tests and various failure scenarios.

The test plan for Network Testing consists of the following tests:

TestID	Host	Action	Expected Outcome
NT4	any	ping every other host	successful ping
NT5	any	pull primary link during continuous ping to any other host	failover to secondary link, no noticeable network interruption
NT6	any	pull standby link during continuous ping to any other host	no effect
NT7	Active L2	pull both network links	Passive L2 becomes Active, L1s fail over to new Active L2
NT8	Passive L2	pull both network links	no effect
NT9	switchA	reload	nodes detect link down and fail to standby link, brief network outage if VRRP transition occurs
NT10	switchB	reload	brief network outage if VRRP transition occurs
NT11	switch	pull single trunk link	no effect

Terracotta Cluster Tests

All tests in this section should be run after the Network Tests succeed.

Test Plan - Active L2 System Loss Tests - verify Passive Takeover

The test plan for Passive takeover consists of the following tests:

TestID	Test	Setup	Steps	Expected Result
TAL1	Active L2 Loss - Kill	L2-A is active, L2-B is passive. All systems are running and available to take traffic.	1. Run app 2. Kill -9 Terracotta PID on L2-A (Active)	L2-B(passive) becomes active. Takes the load. No drop in TPS on Failover.
TAL2	Active L2 Loss - clean shutdown	L2-A is active, L2-B is passive. All systems are running and available to take traffic.	1. Run app 2.Run ~/bin/stop-tc-server.sh on L2-A (Active)	L2-B(passive) becomes active. Takes the load. No drop in TPS on Failover.
TAL3	Active L2 Loss - Power Down	L2-A is Active, L2-B is passive. All systems are running and available to take traffic	1. Run app 2. Power down L2-A (Active)	L2-B(passive) becomes active. Takes the load. No drop in TPS on Failover.
TAL4	Active L2 Loss - Reboot	L2-A is Active, L2-B is passive. All systems are running and available to take traffic	1. Run app 2. Reboot L2-A (Active)	L2-B(passive) becomes active. Takes the load. No drop in TPS on Failover.
TAL5	Active L2 Loss - Pull Plug	L2-A is Active, L2-B is passive. All systems are running and available to take traffic	1. Run app 2. Pull the power cable on L2-A (Active)	L2-B(passive) becomes active. Takes the load. No drop in TPS on Failover.

Test Plan - Passive L2 System Loss Tests

System loss tests confirms High Availability in the event of loss of a single system. This section outlines tests for testing failure of the Terracotta Passive server.

The test plan for testing Terracotta Passive Failures consist of the following tests:

TestID	Test	Setup	Steps	Expected Result
TPL1	Passive L2 loss - kill	L2-A is active, L2-B is passive. All systems are running and available to take traffic.	1. Run app 2. Kill -9 L2-B (Passive)	data directory needs to be cleaned up, then when L2-B is restarted, it re-synchs state from Active Server.
TPL2	Passive L2 loss - clean	L2-A is active, L2-B is passive. All systems are running and available to take traffic	1. Run app 2. Run ~/bin/stop-tc-server.sh on L2-B (passive)	data directory needs to be cleaned up, then when L2-B is restarted, it re-synchs state from Active Server.
TPL3	Passive L2 loss - power down	L2-A is active, L2-B is passive. All systems are running and available to take traffic	1. Run app 2. Power down L2-B (Passive)	data directory needs to be cleaned up, then when L2-B is restarted, it re-synchs state from Active Server.
TPL4	Passive L2 loss - reboot	L2-A is active, L2-B is passive. All systems are running and available to take traffic	1. Run app 2. Reboot L2-B (Passive)	data directory needs to be cleaned up, then when L2-B is restarted, it re-synchs state from Active Server.
TPL5	Passive L2 loss - Pull Plug	L2-A is active, L2-B is passive. All systems are running and available to take traffic	1. Run app 2. Pull plug on L2-B (Passive)	data directory needs to be cleaned up, then when L2-B is restarted, it re-synchs state from Active Server.

Test Plan - Failover/Failback Tests

This section outlines tests to confirm the cluster ability to fail-over to the Passive Terracotta server, and fail back.

The test plan for testing fail over and fail back consists of the following tests:

TestID	Test	Setup	Steps	Expected Result
TFO1	Failover /Failback	L2-A is active, L2-B is passive. All systems are running and available to take traffic	1. Run application 2. Kill -9 (or run stop-tc-server) on L2-A (Active) 3. After L2-B takes over as Active, start-tc-server on L2-A. (L2-A is now passive) 4. Kill -9 (or run stop-tc-server) on L2-B. (L2-A is now Active)	After first failover L2-A->L2-B, txns should continue. L2-A should come up cleanly in passive mode when tc-server is run. When second failover occurs L2-B->L2-A, L2-A should process txns.

Test Plan - Loss of Switch Tests



This test can only be run on a redundant network

This section outlines testing the loss of a switch in a redundant network, and confirming that no interrupt of service occurs.

The test plan for testing failure of a single switch consists of the following tests:

TestID	Test	Setup	Steps	Expected Result
TSL1	Loss of 1 Switch	2 Switches in redundant configuration. L2-A is active, L2-B is passive. All systems are running and available to take traffic.	1. Run application 2. Power down /pull plug on Switch	All traffic transparently moves to switch 2 with no interruptions

Test Plan - Loss of Network Connectivity

This section outlines testing the loss of network connectivity.

The test plan for testing failure of the network consists of the following tests:

TestID	Test	Setup	Steps	Expected Result
TNL1	Loss of NIC wiring (Active)	L2-A is active, L2-B is passive. All systems are running and available to traffic	1. Run application 2. Remove Network Cable on L2-A	All traffic transparently moves to L2-B with no interruptions
TNL2	Loss of NIC wiring (Passive)	L2-A is active, L2-B is passive. All systems are running and available to traffic	1. Run application 2. Remove Network Cable on L2-B	No user impact on cluster

Test Plan - Terracotta Cluster Failure

This section outlines the tests to confirm successful continued operations in the face of Terracotta Cluster failures.

The test plan for testing Terracotta Cluster failures consists of the following tests:

TestID	Test	Setup	Steps	Expected Result
TF1	Process Failure Recovery	L2-A is active, L2-B is passive. All systems are running and available to traffic	1. Run application 2. Bring down all L1s and L2s 3. Start L2s then L1s	Cluster should come up and begin taking txns again
TF2	Server Failure Recovery	L2-A is active, L2-B is passive. All systems are running and available to traffic	1. Run application 2. Power down all machines 3. Start L2s and then L1s	Should be able to run application once all servers are up.

Client Failure Tests

This section outlines tests to confirm successful continued operations in the face of Terracotta client failures.

The test plan for testing Terracotta Client failures consists of the following tests:

TestID	Test	Setup	Steps	Expected Result
TCF1	L1 Failure -	L2-A is active, L2-B is passive. 2 L1s L1-A and L1-B All systems are running and available to traffic	1. Run application 2. kill -9 L1-A.	L1-B should take all incoming traffic. Some timeouts may occur due to txns in process when L1 fails over.