Netframe

ARCHITECTURE

Designing High-Availability Clusters with Netframe

Principles and patterns for building resilient HA clusters, covering network, storage, sizing, and common failure scenarios

May 2026 · Version 1.0

Contents

Overview

High availability in a virtualisation platform is the property that the loss of any single component (a host, a network link, a storage path) does not result in workload outage. On Netframe, HA is enabled on a cluster with a single toggle in Netframe Manager. Once enabled, the cluster monitors host health and, on host failure, restarts the affected VMs automatically on the surviving hosts.

This document is a short, practical overview of how to design HA clusters on Netframe. It is not a configuration manual; the goal is to make sure the right architectural decisions are in place before you build, so the configuration that follows has something solid to stand on.

The four things that matter most are the cluster size, the network topology, the shared storage topology, and the failure scenarios you have explicitly designed for. Each is covered briefly below.

Cluster size and headroom

A meaningful HA cluster needs at least three hosts. Beyond that, the choice of size is a function of redundancy posture and operational tolerance.

N+1. The cluster has one host of spare capacity. The loss of any single host can be absorbed without service impact. This is the minimum that justifies the description “HA cluster” for most production environments. With three hosts, this means each host can run no more than two-thirds of its workload capacity.

N+2. The cluster has two hosts of spare capacity. Suitable for clusters where a host can be down for planned maintenance while still being able to tolerate an unplanned failure of another host. Recommended for clusters above six hosts or where the maintenance window for any single host could extend across days.

Beyond N+2. Rarely justified for an individual cluster; usually it makes more sense to add another cluster and partition workloads across them.

The practical implication is that capacity planning should treat the headroom as committed, not as a buffer. A cluster running at 90% of its host count’s capacity has no real HA, even if the management UI reports green.

Network topology

Two networks matter for HA: the cluster management network (the path Netframe Manager uses to coordinate with Core hosts) and the workload network (the path VM traffic uses to reach the outside world).

The cluster network should be physically redundant. A minimum-credible configuration is two NICs per host, bonded across two physically separate switches. A single switch failure must not partition the cluster.

The workload network has the same redundancy requirement but for a different reason: a VM whose only network path is down is no more available than a VM on a failed host. Bond the workload NICs across the same two switches, on a separate VLAN trunk from the cluster network.

Open vSwitch on each Core host handles both. The bonding mode, MTU, and VLAN trunking are configured per host and visible from Netframe Manager. With cluster-level VM net operations, adding a VLAN once propagates to every host in the cluster.

A common mistake is to consolidate the cluster and workload networks onto the same physical NICs to save on switch ports. This works fine until the workload network is saturated by a misbehaving VM, at which point the cluster network's heartbeat is delayed enough to trigger a spurious HA event. Keep them on separate bonds where possible; if they must share NICs, configure QoS to give the cluster heartbeat priority.

Shared storage topology

Netframe HA requires shared storage. The cluster cannot restart a VM on a different host if the disks for that VM are only accessible from the host that just failed.

Netframe uses NFS as the shared storage backend. The choice of NFS appliance, the network path to it, and the redundancy of that path all matter for HA.

The minimum-credible storage topology has:

- A storage appliance with dual controllers and automatic failover between them.
- Dual storage network paths to each Netframe host, ideally on physically separate switches.
- An NFS export accessible from every host in the cluster, with consistent mount options.

With Netframe Manager's cluster-level batch operations, adding an NFS storage pool is a single action that mounts it consistently across every host in the cluster.

The most common HA gap on the storage side is a single storage switch acting as the single point of failure for shared storage. The HA cluster will report green, the storage appliance will report green, and the loss of that single switch will take down every VM in the cluster simultaneously. Validate the storage path's redundancy with a real test (pull a cable in the right place) before declaring the cluster production-ready.

Failure scenarios to design against

The following failure scenarios should be walked through explicitly during design, with a defined response for each.

Single host failure. Cluster detects the loss and restarts the host's VMs on surviving hosts. Service impact: outage equal to VM restart time, typically tens of seconds per VM. Recovery time: until the failed host is replaced or repaired.

Single network link failure. Bond fails over to the surviving link. Service impact: brief packet loss measured in seconds, no VM outage. Recovery time: until the failed link is replaced.

Single switch failure. Both the cluster network bond and the workload network bond fail over to the surviving switch. Service impact: brief packet loss, no VM outage. Recovery time: until the failed switch is replaced.

Storage appliance controller failure. Storage appliance fails over between controllers. Service impact: brief I/O pause, typically seconds, no VM outage. Recovery time: dual-controller appliance restored to redundant state.

Loss of two hosts simultaneously. With N+1 capacity, surviving hosts may not have room to restart all affected VMs. Some VMs will remain offline until additional capacity is restored. With N+2, the cluster absorbs the loss without service impact.

Patterns we recommend

Three patterns to apply consistently.

Test the failure modes before production. A cluster that has not had a host pulled while running production-like load has not had its HA validated. Set aside a half-day for fault injection during the build phase.

Document the recovery runbook. For each failure scenario, the operations team should know what they will see, what they should do, and what success looks like. The runbook is short, usually a page per scenario, but the existence of it is what makes 2am incidents predictable.

Treat HA as continuous, not as a one-time milestone. Adding capacity, changing the network, upgrading firmware: each of these can erode the HA design without the cluster reporting any change in health. Periodic re-validation, ideally quarterly, catches drift before it matters.

Next steps

If you are designing an HA cluster on Netframe, the practical next steps are:

1. Confirm the cluster size, headroom posture, and storage topology against the principles above.
2. Walk through each of the failure scenarios with the operations team and produce a runbook entry for each.
3. Schedule a fault-injection exercise during the build phase, before any production workload arrives.

For environments where the HA design is critical or unusual, Neon Dynamics can review the proposed architecture before build.