

A modern open source DHCP system from the authors of the classic ISC DHCP

Kea

Key features of Kea include:

- DHCPv4 & DHCPv6
- Dynamic DNS (DDNS)
- Optional lease database backend
- Hooks interface
- REST management API

High Availability Library

Kea Hooks libraries extend the base functionality of the open source Kea DHCP server.

The Kea HA Library is supported with Kea 1.4 and later versions.

Kea extension libraries extend the base functionality of the open source Kea DHCP server. The High Availability Library supports a pair of Kea servers operating in high availability mode, where if one server fails, the other will take over servicing those clients.

Features

Supports pairs of either DHCPv4 or DHCPv6 servers.

Active leases are continuously updated between the HA peers.

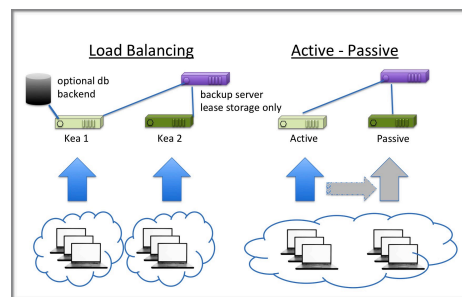
HA peers may be using either the default memfile backend or an optional database backend for lease and reservation storage.

High Availability

Since DHCP is a critical network service, many operators deploy DHCP servers in pairs for redundancy. The Kea HA hook library adds a heartbeat between the paired servers. When one server observes the heartbeat has failed, it can take over responding to clients associated with its peer. Because both servers are continuously sending each other updates with their current lease state, clients can renew existing leases with the partner server without delay or interruption.

Load-balancing and Active-passive modes

The Kea HA feature works in load-balancing mode, with client responsibilities split 50/50 between peers. Clients broadcast their DHCPDISCOVER request and each server determines independently if it is responsible for that client using Pearson's Hash, described in IETF RFC 3074. Kea HA also supports an active-passive (aka "standby")



mode, with client responsibilities split 100/0 during normal operation. In this case, when the active server fails, the standby takes over serving all leases. Lease updates may also optionally be replicated to an additional backup server that is not an active DHCP server, for secure record-keeping.

Static configuration

To use the Kea HA feature, configure all available addresses in both servers, with pools, subnets and shared networks assigned to the server that will be primarily responsible for managing them. (see example configuration below)

In deployments where peers use a shared database backend for central storage of the host reservation data either peer will be able to manage any static host reservation.

FREQUENTLY ASKED QUESTIONS

Example configuration showing a subnet split across a pair of servers

The following subnet configuration includes two pools. One of the pools is only allowed for packets classified as belonging to "server1", another one is only allowed for packets belonging to "server2". When the servers receive the client request, they evaluate the load balancing algorithm and then check to see if the value specified in *this-server-name* matches the one configured for a pool on that server.

```
"subnet4": [
  {
    "subnet": "192.0.2.0/24", "pools": [
      {
        "pool": "192.0.2.1 - 192.0.2.50",
        "client-class": "server1"
      },
      {
        "pool": "192.0.2.51 - 192.0.2.100",
        "client-class": "server2"
      }
    ]
  }
]
```

Similarly, it is possible to split subnets rather than pools to different HA peers. In such case, the client classes should be specified in the subnet scope.

Comparison with Failover in ISC DHCP

There are a few key differences between the Kea HA feature and the implementation of IETF standard failover in ISC DHCP. The Kea HA feature is *not* the DHCPv4 failover feature specified in the IETF – it is a lightweight, proprietary feature that leverages Kea's existing control channel and lease commands hook library. Unlike the IETF standard feature, Kea HA also works (exactly the same way) for both DHCPv4 and DHCPv6.

Kea does not handle time synchronization between servers, address ranges must be statically divided between servers, and Kea does not support 'ping before use' to check for address conflicts, which ISC DHCP does. For a full comparison see: <https://kb.isc.org/article/AA-01617/210/Kea-High-Availability-vs-ISC-DHCP-Failover.html>

Can I use the HA feature if my Kea instance is using a database backend?

Yes. In this configuration the leases will be stored in the database backend and synchronized between the database instances via whatever database clustering mechanism you have deployed. Kea can send heartbeats over the control channel between two Kea servers and trigger the partner-down state even though the Kea HA is not managing lease updates.

How can I get the High Availability Library?

The HA library part of the Kea open source distribution and is available for download from the ISC web site (www.isc.org)