Tomek Mrugalski

tomek(at)isc(dot)org

UKNOF40

A modern DHCP engine

KEA
If you never heard about Kea…

- 1.4.0 beta about to be released (May 2018)
- Open source (MPL2)
- Linux, BSDs, MacOS, ...
- REST Management API
- Hooks (3rd party libraries, like Apache modules)
- Database Backends
- Feature rich: shared networks, v6, PD, custom options’…
- On-line reconfiguration (no restarts after config changes)
- Scalable (millions of devices)
- Performance
- Modern DHCPv4, DHCPv6 and DDNS servers
Leases, host reservations in DB (1.4)

- Use host commands (1.2) and subjects (1.3)
- Can manipulate the DB directly or configuration in DB likely in 1.5
- SQL data can be modified any time
- All changes applied instantly (no restart)
- Cassandra, MySQL, PostgreSQL, CSV
- Can manipulate the DB directly
Hooks (1 of 2)

1.1: User Check
- example access control

1.2: Forensic Logging
- audit trail for legal purposes

1.2: Flexible Identifier
- identify hosts by expression,
  e.g. concat(relay4[2].hex, relay4[6].hex)

1.2: Host Commands
- query, add and delete host

1.3: Subnet management
- add and delete network, subnet and shared networks via REST API
- extra lease commands (add, get, update, delete)

1.3: Extra lease commands
- add all, get all leases via REST API

1.3: Subnet management
- add and delete host reservation using REST interface

1.1: User Check
- example access control

Open source
Premium
1.4: **Radius** – access control and host reservation

- detection, lease updates, recovering DB from partner

1.5: **Limits** - ability to rate limit queries, limit # of

(includes negative caching)

(heartbeats, failure)

1.4: **Host Caching** – cache host responses locally

using FreeRadius, accounting

Hooks (2 of 2)
Anyone can write hooks

Photo by Angelo Failla, Facebook
### How to identify hosts:

- **Open source**
  - MAC, duid, circuit-id, client-id

- **Almost anything could be used** (35 different expressions)
  - Options (client, relay, vendor)
  - Fixed fields
    - Concat, substring
    - Options (client, relay, vendor)
  - Meta-data (interface name, ...

- **Premium**
  - MAC, duid, circuit-id, client-id
  - Open source
  - How to identify hosts:

### Meta-data

- Interface name
- src/dst IP
- ...
REST API (1.2/1.3)

Overview:

- Command Channel (Unix socket)
- REST interface (http/https)
- JSON commands, JSON responses
- kea-shell provided (python 2.x, 3.x example)
- More to come in future releases

Manipulate:

- Whole config (config-get/set/test/write)
- Shared networks, subnets (subnet4/6-list/add/get/del)
- Leases (lease-*get/addr/update/del/wipe)
- Statistics (statistic-get/reset/get-all)
- Server (list-commands, shutdown, version-get, build-report, leases-reclaim, etc.)

Response

```
{
  [ {  
    "subnet": "2001:db8:1::/64",
    "Td": 234,  
    } ]
  "arguments": "subnet already added",
  "text": "IPv6 subnet added",
  "Result": 0 }
```

Command

```
{
  ... [ {  
    "subnet": "2001:db8:1::/64",
    "Td": 234,  
    } ]
  "arguments": "subnet already added",
  "command": "subnet-add" }
```
High Availability (1.4)

- Load balancing or hot standby
- Backup server database
- Auto-sync of lease
- Failure detection based on 'secs' field
- Lease updates via lease_cmds hook
- Bi-directional lease updates
- 50/50 LB split
- 50% of leases assigned & managed by each server
- Heartbeats over control channel
- Hook points (sending lease updates, etc.)
- RESTful API based
- Failover or hot standby
- V4 and V6
- Kea server 1
- Kea server 2
- 50%
Apache Cassandra (1.4)

- Distributed non-relational NoSQL database
- Massive scalability without a single point of failure
- Replication factor
- Can operate with at least one node surviving
- CQL
- Data denormalization
- RF = 2N + 1 can survive N failures

<table>
<thead>
<tr>
<th>RF</th>
<th>NODES</th>
<th>FAILURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

```
RF = 2N + 1 can survive N failures
```
1.4 coming up

- 1.4.0 beta: May 14th, 1.4.0 final: June 15th
- Improved shared networks performance
- Improved classification
- Fixed statistics when run multiple instances with the same DB
- member(foo) && !member(bar) && (relay4[2].hex == 'abcd')
- Many smaller bugfixes and improvements
- (100+ tickets closed and counting)
DHCPv4 and DHCPv6
FreeRadius based
Attributes customizable
Accounting
Class assignment
Address Reservation
Access control
Authentication

RADVUS Integrateation (1.4)
Kea. isc.org/wiki/centralizedConfiguration

DB Configuration Storage (1.5)
Yang/Netconf (1.5)

YANG model

```yaml
{,
    description "This model supports a hierarchy;",
    container network-ranges
        {,
            description "For IA",
            mandatory true;
            type yang:timeticks;
        }
    network-ranges
        {,
            description "A container that describes the",
            mandatory true;
            leaf network-range-id
                {,
                    description "Identifies the network-range;",
                    type uint32;
                }
            network-range
                {,
                    description "This model supports a hierarchy;",
                    mandatory true;
                    leaf network-range-id
                        {,
                            description "Identifies the network-range;",
                            type uint32;
                        }
                    network-range
                        {,
                            description "A container that describes the",
                            mandatory true;
                            leaf network-range-id
                                {,
                                    description "Identifies the network-range;",
                                    type uint32;
                                }
                            network-range
                                {,
                                    description "For IA",
                                    mandatory true;
                                    type yang:timeticks;
                                }
                        }
                }
        }
}
```

Configuration

```xml
<server>
    <server-config>
        <network-ranges>
            <network-range>
                <network-range-id>1</network-range-id>
                <network-description />
                <network-prefix>2001:db8::/56</network-prefix>
            </network-range>
            <address-pools>
                <address-pool>
                    <pool-id>1</pool-id>
                    <start-address>2001:db8::1</start-address>
                    <end-address>2001:db8::ffff</end-address>
                    <renew-time>20</renew-time>
                    <rebind-time>90</rebind-time>
                    <valid-lifetime>150</valid-lifetime>
                    <preferred-lifetime>120</preferred-lifetime>
                </address-pool>
            </address-pools>
        </network-ranges>
    </server-config>
</server>
```
TreeView for YANG model
Useful Links

- Source code for premium hooks is also provided to purchasers: https://github.com/isc-projects/kea
- Kea comparison, support links, 24/7 support available.
- High level overview, premium hooks white papers, ISC DHCP vs Kea.
- Kea project homepage: https://isc.org/kea
- Kea business page: https://isc.org/kea
- Developer’s Guide - for developers and contributors, explains the internals, also includes Hooks interface API.
- Why a nod towards the mainframe era.
- List of all log messages - with an explanation what happened and hooks (easy to see if you would benefit from purchasing them).
- REST API documentation, and user documentation for premium.
- Users Guide - 100+ pages of guidance with examples for users.
- Documentation: https://kea.isc.org/docs/
- Kea project homepage: https://kea.isc.org/kea
Tomatoes?
Suggestions?
Questions?

Q&A