Kea – modern DHCP server
Open Source and Sustainability

Tomek Mrugalski
tomasz@isc.org
What is Kea?

And why you may want to use it
If you never heard about Kea…

- Modern DHCPv4 and DHCPv6 server (1.0 in Dec 2015)
- Performance (1000s leases/sec)
- Scalable (millions of devices)
- No restarts after config changes
- Databases (CSV, MySQL, PostgreSQL, Cassandra)
- Hooks (3rd party libraries)
- REST management API
- Linux, BSDs, MacOS, …
- Open source (MPL2)
- 1.2.0 just released (28 April 2017)
Let’s compare!

<table>
<thead>
<tr>
<th></th>
<th>ISC DHCP</th>
<th>ISC Kea</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Not adding anything big</td>
<td>Active development with tons of new features</td>
</tr>
<tr>
<td>Code repository</td>
<td>Internal, tarball published</td>
<td>github</td>
</tr>
<tr>
<td>Bug database</td>
<td>Internal, mail external</td>
<td>public trac</td>
</tr>
<tr>
<td>Testing</td>
<td>~30 unit-tests</td>
<td>4000+ unit-tests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Memory leak tests (valgrind)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>700+ system tests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuzz testing</td>
</tr>
<tr>
<td>Docs</td>
<td>Man pages</td>
<td>User’s Guide (100+ pages)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developer’s Guide</td>
</tr>
<tr>
<td>Logs</td>
<td>Fixed log message</td>
<td>Every possible log entry is documented and described</td>
</tr>
<tr>
<td>IPv6 readiness</td>
<td>IPv4 originally, IPv6 added</td>
<td>IPv4 optional</td>
</tr>
<tr>
<td></td>
<td>later</td>
<td></td>
</tr>
</tbody>
</table>
# Why migrate from ISC DHCP?

<table>
<thead>
<tr>
<th></th>
<th>ISC DHCP</th>
<th>ISC Kea</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td>Ok (with ramdisk tricks)</td>
<td>Great (many 1000s leases/sec)</td>
</tr>
<tr>
<td><strong>Management</strong></td>
<td>OMAPI (custom C interface)</td>
<td>JSON over REST API/http, JSON over Unix socket</td>
</tr>
<tr>
<td><strong>Extensibility</strong></td>
<td>Shell scripts (out only),</td>
<td>JSON everywhere, Hooks (C++), stable API</td>
</tr>
<tr>
<td></td>
<td>configuration language</td>
<td></td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td>Custom complex syntax (almost</td>
<td>JSON with optional DB storage for some elements (more to</td>
</tr>
<tr>
<td></td>
<td>programming language)</td>
<td>come)</td>
</tr>
<tr>
<td><strong>Leases information</strong></td>
<td>Custom</td>
<td>CSV, MySQL, PgSQL, Cassandra</td>
</tr>
<tr>
<td><strong>Hosts information</strong></td>
<td>Custom config</td>
<td>JSON, MySQL, PgSQL</td>
</tr>
</tbody>
</table>
Cool features :: DB

- Leases, host reservations in DB
  - CSV
  - MySQL or PostgreSQL
  - Cassandra*
- SQL data can be modified any time
- All changes applied instantly (no restart)
- Can fiddle with the DB directly or
- Use host commands (1.2) or subnets (1.3)
Cool features :: REST

- Command Channel (Unix socket, since 0.9.2)
- REST interface (http, since 1.2.0)
- JSON commands, JSON responses
  - kea-shell provided (python 2.x, 3.x example)
  - Trivial to use from any JSON/http capable env
- Commands:
  - config-get, config-set, config-test, config-write
  - reservation-get, reservation-add, reservation-del*
  - statistic-get, statistic-reset, statistic-get-all, statistic-reset-all, ...
  - leases-reclaim, list-commands, shutdown, version-get, build-report
- More to come every release
Kea Roadmap

- REST interface
- Rewritten configuration handling
- Commands (config-set/get/test/write)

1.2
(Apr 2017)

Host Commands
Flexible Identifier

1.3
(Oct 2017)

Subnet Commands

- Shared subnets
- Security for REST interface
- Lease commands
- ...

1.4
(spring 2018)

Multi-proc support
Better High Availability/Redundancy
DB improvements
YOUR FEATURE HERE

TBD

Open source
Premium

Tomek Mrugalski, 2017-05-11
Cool features :: Hooks

Facebook datacenter running on Kea

Photo by Angelo Failla, Facebook
Open Source and Sustainability

How to properly fund OS?
Commercial quality software

- A small team (2 full time, with 2 more contributing) of experienced engineers
- A real, independent QA
  - 4000+ unit-tests, 700+ system tests
  - Run on ~20 systems
  - Valgrind, Coverity scan, other static analyzers
- Proper designs
  - Written Requirements, Designs, Implementation, Testing
- Very well documented
  - User’s Guide (100+ pages, ~40 example configs)
  - Developer’s Guide (code is well commented, all params documented)
Funding so far

- In development since 2011
- Had several custom development contracts
- Two sponsors (Comcast and Mozilla, thanks!)
- Very few support customers
- Sporadic personal contributions (thanks!)

But...

Most of the work was internally funded by ISC
Funding Idea #1: Kea 1.2.0 Docker

- Docker image with Kea + MySQL pre-configured
- Easy to deploy
- An experiment
Funding Idea #2 : Premium features

- Kea is and will remain open source (MPL2, 478KLOC)
  - Provides support for hook libraries (~Apache module)
  - API is open (3rd party hook libs appearing now)

- Premium (EULA, 6.8KLOC)
  - Additional extra features
  - targeted for large deployments
  - A way to convince people to sign support contract
  - Yes, support contract = $

isc.org/blogs/funding-kea/
Existing & Planned Hooks

1.2

- **User_chk** – example access control (open source)
- **Forensic Logging** – detailed audit trail for legal purposes
- **Flexible Identifier** – identify hosts by expression, e.g. `concat(relay4[2].hex, relay4[6].hex)`
- **Host Commands** – query, add and delete host reservations using REST interface

1.3

- **Subnet management** (add, get, update, delete)
- **Extra lease commands** (add, get, update, delete)
Hook Example

Flexible Identifier

How to identify hosts:

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

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

concat(relay4[1].hex, relay4[2].hex)
Funding Idea #3: Kittiwake GUI

- REST interface
- Focusing on typical ops first:
  - Monitor pool utilization
  - Add/delete host reservations
- Distribution/Licensing TBD
  - Freemium?
  - Paid app?
  - Extra benefit for customers?
- 1.0 planned in Autumn 2017
- Get in touch! We’ll be looking for beta testers/your requirements
Funding Idea #3: Kittiwake GUI

- REST interface
- Focusing on typical ops first:
  - Monitor pool utilization
  - Add/delete host reservations
- Distribution/Licensing TBD
- Freemium?
- Paid app?
- Extra benefit for customers?
- 1.0 planned in Autumn 2017

Get in touch! We’ll be looking for beta testers/your requirements.
Funding Idea #4: ISC DHCP to Kea migration

- Migration tool is in development
- ISC DHCP config is complex (80% cases)
- Trials will start soon
- Interested?
  - We’re looking for configuration samples
  - Talk to us
- Revenue model TBD
Your thoughts on those ideas?

Questions?

Suggestions?