Kea Webinar

Installation and Configuration

Carsten Strotmann

30th September 2020

https://www.isc.org

Welcome

- Welcome to part two of our webinar series "the KEA DHCP Server"

About this Webinar

- What is Kea DHCP
- Installation, Platform and Requirements
- Kea Hooks
- Kea basic configuration
- Testing the Kea DHCP server function
- Operating the Kea server
- Kea DHCPv6

What is Kea DHCP?

Kea DHCP (1/2)

- a modern DHCPv4 and DHCPv6 Server
- Open Source (MPL License)
- JSON/REST API
- modular design
- high performance (> 1000 leases/seconds with SQL database backend)
- failover via SQL DB or High-Availability-Hook
- Host reservation support

Kea DHCP (2/2)

- support for DHCPv6 prefix delegation
- dynamic reconfiguration
- dynamic DNS updates
- SQL database backend (MySQL / PostgreSQL / Cassandra …)
- Statistics module
- PXE Boot support

Platforms and Requirements

Kea Platforms (1/2)

- Kea is officially supported on
  - CentOS Linux
  - Fedora Linux
  - Ubuntu Linux
  - Debian Linux
  - FreeBSD Unix

Kea Platforms (2/2)

- Kea is also know to work on
  - MacOS X
  - Alpine Linux
  - OpenBSD Unix
- Kea currently does not work on Windows operating systems

Kea DHCP requirements

- Kea requires to run
  - a cryptographic library: Botan or OpenSSL
  - log4cplus C++ logging library
  - the Boost C++ system library
- optional components
  - a database such as MySQL, PostgreSQL or Cassandra (CQL)
  - FreeRadius-client library for Radius support
  - Sysrepo for NETCONF support
- see the Kea DHCP documentation for detailed information on software dependencies

Installation

Kea DHCP installation via operating system packages

- Kea DHCP is available in the package repositories of all major Linux and Unix systems
  - If you have support from the operating system vendor (Red Hat, Canonical, Suse), installing from the OS repositories is the best choice
- Kea DHCP can also be installed from source, if you need a special build configuration or the latest features not available in the binary packages

Kea DHCP installation via ISC packages

- ISC offers binary packages of Kea DHCP for our users and customers, hosted on Cloudsmith.
  - if you need the latest Kea version, these packages are an alternative to building Kea from source
- the packages provide fast access to the latest bug fixes
  - ISC provides the binary packages along with sources at the time of release

Kea binary packages from ISC

- The open source packages contain the base Kea software and the following hooks libraries:
  - Flexible Option
  - Lease Commands
  - High Availability
  - Statistics Commands
  - BOOTP

Packages for support customers

- Users of Kea that purchase professional Kea DHCP support from ISC are entitled to special software features that are not available in the open source version:
  - Class Commands
  - Configuration Backend Commands
  - Flexible Identifier
  - Forensic Logging
  - Host Cache
  - Host Commands
  - RADIUS support
  - Subnet Commands

Kea hooks

The Kea hooks

- The base Kea software implements the basic DHCPv4 and DHCPv6 functions
- These basic functions can be extended via hooks.
  - The hooks are libraries that contain extra functions that will be called when a DHCP request is processed
  - Hooks allow the core Kea system to stay lean
  - Installations only load the functions used and needed
  - This reduces the complexity and the attack surface of an installation

Types of hooks available

- Hooks that are part of the Kea open source code (source and binary packages)
- Premium hooks that can be purchased online from the ISC website
- Hooks that are available for ISC support subscription customers
- Third party hooks (source code)

Premium/Subscription hooks

- The premium/subscription hooks are available in source and binary (package) form
  - Customers can download the hooks for a period of 12 month
  - As the API between Kea and the hooks might change between Kea versions, care must be taken to install hooks that match the Kea version number

Kea configuration

JSON

- Configuration files for the **DHCPv4**, **DHCPv6**, **DDNS**, **Control Agent**, and **NETCONF** modules are defined in an extended JSON format.
- Basic JSON is defined in RFC 7159 and ECMA 404.

```
{
"Dhcp4": {
    "interfaces-config": {
        "interfaces": [ "eth0" ]
    },
    "control-socket": {
        "socket-type": "unix",
        "socket-name": "/tmp/kea-dhcp4.socket"
    },
[...]
```

Extended JSON

- Kea components use an extended JSON with additional features:
  - Shell comments: any text after the hash (#) character is ignored.
  - C comments: any text after double slashes (//) is ignored.
  - Multiline comments: any text between /* and */ is ignored. This commenting can span multiple lines.
  - File inclusion: JSON files can include other JSON files by using a statement of the form `<?include "file.json"?>`.

JSON Editor

- When working with KEA, it helps to have an editor that understands the JSON format, can check the syntax and can highlight and reformat JSON data
  - Emacs
  - VIM
  - Visual Studio Code
  - TextMate / BBEdit
  - …

EMACS JSON Mode

- EMACS JSON Mode: https://www.emacswiki.org/emacs/JSON
  - Enable JSON-Mode in Emacs with

    `ESC-X json-mode<enter>`
  - Re-format a JSON file with `CTRL+c-CTRL+f`

VIM JSON Syntax Highlighting

- Using VIM, syntax highlighting for JSON can be enabled in the command mode with

```
set syntax=json
```

Location of the KEA configuration files

- On most Linux/Unix systems, the Kea configuration files can be found under `/etc/kea/`
  - some have their own locations such as `/usr/local/etc/kea` on FreeBSD

Kea configuration files

- The main Kea configuration file are
  - `kea-ctrl-agent.conf` - Kea control agent
  - `kea-dhcp-ddns.conf` - Kea dynamic DNS updater
  - `kea-dhcp4.conf` - Kea DHCPv4 server
  - `kea-dhcp6.conf` - Kea DHCPv6 server
  - `keactrl.conf` - configuration file for keactrl script (not in JSON format)

Documentation

- The example configuration files provided by the Kea project contain extensive comments
- The full documentation can be found online at `https://kea.readthedocs.io`

A basic Kea DHCPv4 configuration

Network Interface and control socket

- The Kea DHCP server needs to know on which network interfaces the DHCP service should listen on
- The control socket defines the communication interface between the DHCP server process and the administration tools

```
{
"Dhcp4": {
    "interfaces-config": {
        "interfaces": [ "eth0" ]
    },
    "control-socket": {
        "socket-type": "unix",
        "socket-name": "/tmp/kea-dhcp4.socket"
    },
[...]
```

Lease database definition

- Kea DHCP needs to know where to store the lease information. The configuration snippet below defines a in-memory database

```
[...]
    "lease-database": {
        "type": "memfile",
        "lfc-interval": 3600
    },
[...]
```

Global configuration and options

- Some DHCP configurations are global and apply to all the subnets and pools managed by the DHCP server

```
[...]
    "renew-timer": 900,
    "rebind-timer": 1800,
    "valid-lifetime": 3600,
[...]
```

IPv4-Subnet and Pool definition

- The example of a subnet with DHCP pool definition includes subnet specific options (default router option: routers)

```
[...]
    "subnet4": [
        {
            "subnet": "192.0.2.0/24",
            "pools": [ { "pool": "192.0.2.100 - 192.0.2.200" } ],
            "option-data": [
                {
                    "name": "routers",
                    "data": "192.0.2.1"
                }
            ]
        }
    ]
},
[...]
```

Logging

- Kea DHCP has comes with a flexible and powerful logging framework
- The configuration snippet on the right configures a log-file for the DHCPv4 service

```
[...]
"Logging":
{
  "loggers": [
    {
        "name": "kea-dhcp4",
        "output_options": [
            {
                "output": "/var/log/kea-dhcp4.log"
            }
        ],
        "severity": "INFO",
        "debuglevel": 0
    }
  ]
}
}
```

Checking the configuration for syntax errors

Kea configuration syntax check

- After changes to a configuration file, and before reloading the new configuration into the Kea server, the configuration file should be checked for errors
  - syntax checks can be done with the -t (test) parameter

Kea configuration check examples (1/2)

- KEA-DHCP4 Syntax Check with error (line 33, char 9)

```
# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
Syntax check failed with: /etc/kea/kea-dhcp4.conf:33.9: syntax error, unexpected }
```

Kea configuration check examples (2/2)

- Successful Syntax check (output wrapped for readability)

```
# kea-dhcp4 -t /etc/kea/kea-dhcp4.conf
INFO  [kea-dhcp4.dhcpsrv/51] DHCPSRV_CFGMGR_ADD_IFACE listening on interface server-eth0
INFO  [kea-dhcp4.dhcpsrv/51] DHCPSRV_CFGMGR_SOCKET_TYPE_DEFAULT "dhcp-socket-type" not specified,
                             using default socket type raw
INFO  [kea-dhcp4.dhcpsrv/51] DHCPSRV_CFGMGR_NEW_SUBNET4 a new subnet has been added to configuration:
                             192.0.2.0/24 with params: t1=900, t2=1800, valid-lifetime=3600
```

Starting, restarting and stopping Kea

keactrl

- keactrl is a shell script that can be used to control the Kea services
- Care must be taken not to conflict with process supervision services such as systemd, runit or s6

keactrl configuration (1/2)

- The configuration for keactrl lists the location of the configuration files and the Kea binaries

```
# This is a configuration file for keactrl script which controls
# the startup, shutdown, reconfiguration and gathering the status
# of the Kea's processes.

# Note that control agent must be launched after servers and netconf last.

# prefix holds the location where the Kea is installed.
prefix=/usr/local

# Location of Kea configuration files.
kea_dhcp4_config_file=${prefix}/etc/kea/kea-dhcp4.conf
kea_dhcp6_config_file=${prefix}/etc/kea/kea-dhcp6.conf
kea_dhcp_ddns_config_file=${prefix}/etc/kea/kea-dhcp-ddns.conf
kea_ctrl_agent_config_file=${prefix}/etc/kea/kea-ctrl-agent.conf
kea_netconf_config_file=${prefix}/etc/kea/kea-netconf.conf

# Location of Kea binaries.
exec_prefix=${prefix}
dhcp4_srv=${exec_prefix}/sbin/kea-dhcp4
dhcp6_srv=${exec_prefix}/sbin/kea-dhcp6
dhcp_ddns_srv=${exec_prefix}/sbin/kea-dhcp-ddns
ctrl_agent_srv=${exec_prefix}/sbin/kea-ctrl-agent
netconf_srv=${exec_prefix}/sbin/kea-netconf
[...]
```

keactrl configuration (2/2)

- The last part of the configuration defines which Kea services should be started
  - and controls verbose logging

```
[...]
# Start DHCPv4 server?
dhcp4=yes

# Start DHCPv6 server?
dhcp6=yes

# Start DHCP DDNS server?
dhcp_ddns=no

# Start Control Agent?
ctrl_agent=yes

# Start Netconf?
netconf=no

# Be verbose?
kea_verbose=no
```

start Kea services with keactrl

- After creating a configuration for keactrl, the script can be used to start the Kea server modules

```
# keactrl start
INFO/keactrl: Starting /opt/kea/sbin/kea-dhcp4 -c /opt/kea/etc/kea/kea-dhcp4.conf
INFO/keactrl: Starting /opt/kea/sbin/kea-ctrl-agent -c /opt/kea/etc/kea/kea-ctrl-agent.conf
```

Kea service status (keactrl)

- keactrl offers a status overview of the currently configured modules

```
# keactrl status

DHCPv4 server: active
DHCPv6 server: inactive
DHCP DDNS: inactive
Control Agent: active
Kea DHCPv4 configuration file: /opt/kea/etc/kea/kea-dhcp4.conf
Kea DHCPv6 configuration file: /opt/kea/etc/kea/kea-dhcp6.conf
Kea DHCP DDNS configuration file: /opt/kea/etc/kea/kea-dhcp-ddns.conf
Kea Control Agent configuration file: /opt/kea/etc/kea/kea-ctrl-agent.conf
keactrl configuration file: /opt/kea/etc/kea/keactrl.conf
```

reloading Kea services with keactrl

- After changing a Kea configuration file (and checking for errors), keactrl can be used to reload the configuration into the Kea processes

```
# keactrl reload
INFO/keactrl: Reloading kea-dhcp4...
INFO/keactrl: Reloading kea-ctrl-agent...
```

stop Kea services with keactrl

- keactrl can also be used to stop all configured Kea modules

```
# keactrl stop
INFO/keactrl: Stopping kea-dhcp4...
INFO/keactrl: kea-dhcp6 isn't running.
INFO/keactrl: kea-dhcp-ddns isn't running.
INFO/keactrl: Stopping kea-ctrl-agent...
```

start KEA DHCPv4 module via systemd

- On Linux systems, Kea comes with a set of systemd unit files that control the Kea services

```
# systemctl start kea-dhcp4
```

start KEA DHCPv4 module via systemd

- check the status of the Kea DHCPv4 service (Linux systemd)

```
# systemctl status kea-dhcp4
● kea-dhcp4.service - Kea DHCPv4 Server
   Loaded: loaded (/usr/lib/systemd/system/kea-dhcp4.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2018-12-06 10:13:26 UTC; 4s ago
     Docs: man:kea-dhcp4(8)
 Main PID: 63 (kea-dhcp4)
    Tasks: 1 (limit: 1144)
   Memory: 1.9M
   CGroup: /machine.slice/libpod-2e3e4a67333cf94630baa9c268ae84f8e77353abf14b074ed2ef9d73bc6e4f53.scope/system.slice/kea-dhcp4.service
           └─63 /usr/sbin/kea-dhcp4 -c /etc/kea/kea-dhcp4.conf

Dec 06 10:13:26 2e3e4a67333c systemd[1]: Started Kea DHCPv4 Server.
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.674 INFO  [kea-dhcp4.dhcp4/63] DHCP4_STARTING Kea DHCPv4 server version 1.3.0 starting
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.676 INFO  [kea-dhcp4.dhcpsrv/63] DHCPSRV_CFGMGR_ADD_IFACE listening on interface server-eth0
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.676 INFO  [kea-dhcp4.dhcpsrv/63] DHCPSRV_CFGMGR_SOCKET_TYPE_DEFAULT "dhcp-socket-type" not specified , using default socke>
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.677 INFO  [kea-dhcp4.dhcpsrv/63] DHCPSRV_CFGMGR_NEW_SUBNET4 a new subnet has been added to configuration: 192.0.2.0/24 wit>
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.677 INFO  [kea-dhcp4.dhcp4/63] DHCP4_CONFIG_COMPLETE DHCPv4 server has completed configuration: added IPv4 subnets: 1; DDN>
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.677 INFO  [kea-dhcp4.dhcpsrv/63] DHCPSRV_MEMFILE_DB opening memory file lease database: lfc-interval=3600 type=memfile uni>
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.679 INFO  [kea-dhcp4.dhcpsrv/63] DHCPSRV_MEMFILE_LEASE_FILE_LOAD loading leases from file /var/lib/kea/kea-leases4.csv
Dec 06 10:13:26 2e3e4a67333c kea-dhcp4[63]: 2018-12-06 10:13:26.680 INFO  [kea-dhcp4.dhcpsrv/63] DHCPSRV_MEMFILE_LFC_SETUP setting up the Lease File Cleanup interval to 3600 sec
```

Testing DHCPv4 with the ISC dhcp client

ISC dhclient as a debugging tool

- Most Linux distributions provide the ISC DHCP client tool dhclient
- This tool can be used as an simple DHCP debugging tool

dhclient as a debugging tool (1/2)

- Create a new shell script in
  `/usr/local/sbin/dhclient-debug.sh`
  with the lines below

```
#!/bin/sh
env
```

  - this script will print all variables in it's execution environment
- make the script executeable

dhclient as a debugging tool (2/2)

- Execute the dhclient tool with this script

```
dhclient -sf /usr/local/sbin/dhclient-debug.sh
```

- The script will print out all the information received from the DHCP server (via environment variables)
- It will not reconfigure the client machines network stack!

Performance benchmarking: perfdhcp

- Kea comes with a DHCP benchmarking tool: perfdhcp
- This tool can be used to benchmark Kea, but also other DHCP server systems
- For details, see the perfdhcp documentation

`https://kea.readthedocs.io/en/kea-1.6.2/man/perfdhcp.8.html`

Kea control agent

Kea control agent

- The Kea control agent is a process that provides a HTTP(s) REST interface
- The control agent can be used to dynamically reconfigure the Kea services (without manually changing the configuration files)
- The kea control agent communicates with the running Kea services via unix control sockets

Configuration of the Kea control agent

- By default, the Kea control agent listens on the (first) IPv4 loopback address 127.0.0.1 Port 8000
- This can be changed in the configuration file `kea-ctrl-agent.conf`

Kea shell

- The Kea Shell is a Python command line tool to interact with the Kea Control Agent REST API

Kea shell example

- The Kea shell returns the JSON data from the Kea-Modules REST API
  - Tools such as jq can be used to pretty print the output

```
# kea-shell --service dhcp4 --host 127.0.0.1 --port 8000 version-get | jq
[
  {
    "arguments": {
      "extended": "1.7.10-git\ngit 9bade6ae294f570976e7614e84a76a34ac4915b1\nlinked with:\nlog4cplus
1.2.2\nLibreSSL 3.2.1\ndatabase:\nPostgreSQL backend 6.1, library 120003\
nMemfile backend 2.1"
    },
    "result": 0,
    "text": "1.7.10-git"
  }
]
```

Reading configuration data

- The REST interface has been designed to be used from a Kea configuration application (such as Kea Stork or Kea Shell)
- However, API calls can be send to the Kea control agent from the command line via the curl tool
- Here we send the config-get command to the DHCPv4 server

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "config-get", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/
```

Pretty printing the JSON output

- The output is unformatted JSON.
  - The tool jq can be used to pretty-print the output

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "config-get", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/ | jq
```

JSON queries with jq

- jq can be used to filter specific parts of the configuration. The jq filter ".[0].arguments" can be used to produce a valid KEA configuration file.
- The example below prints the logging config of the DHCPv4 server:

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "config-get", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/ | jq ".[0].arguments.Dhcp4.loggers"
```

JSON queries with jq

- Result:

```
[
  {
    "debuglevel": 0,
    "name": "kea-dhcp4",
    "output_options": [
      {
        "output": "/opt/kea/var/log/kea-dhcp4.log"
      }
    ],
    "severity": "INFO"
  }
]
```

List API commands

- The list-commands command returns the API commands available for a specific KEA module

```
[kea-server]# curl -X POST -H "Content-Type: application/json" \
  -d '{ "command": "list-commands", "service": [ "dhcp4" ] }' \
  http://127.0.0.1:8000/ | jq
```

Dynamic changes to the Kea configuration file (1/5)

- With the REST API, it is possible to
  - remotely fetch the current running config of a Kea server
  - change the config
  - and write the config back to the server

Dynamic changes to the Kea configuration file (2/5)

- Dump the current configuration into a file

```
curl -s -X POST -H "Content-Type: application/json" \
     -d '{ "command": "config-get",  "service": [ "dhcp4" ] }' \
     http://127.0.0.1:8000/ | jq ".[0]" > kea-dhcp4.tmp
```

Dynamic changes to the Kea configuration file (3/5)

- Edit the file
  - Add
    the command and service information
  - Make changes to the configuration
  - Remove the result from the JSON file

```
{
    "command": "config-set",
    "service": [ "dhcp4" ],
    "arguments": {
        "Logging": {
            "loggers": [
                {
                    "severity": "INFO",
                    "output_options": [
[...]
```

Dynamic changes to the Kea configuration file (4/5)

- Send the new configuration to the server

```
[kea-server]# curl -s -X POST -H "Content-Type: application/json" \
   -d @kea-dhcp4.tmp http://127.0.0.1:8000/ | jq
[
  {
    "result": 0,
    "text": "Configuration successful."
  }
]
```

Dynamic changes to the Kea configuration file (5/5)

- All dynamic changes are stored in memory
  - To make the changes persistent, write the in-memory configuration back to a file with the config-write command (be careful, any comments in the file will be gone and the formatting will be different)

```
[kea-server]# curl -s -X POST -H "Content-Type: application/json" \
                  -d '{ "command": "config-write", "arguments": { "filename": "/etc/kea/kea-dhcp4-new.json" }, "service": [ "dhcp4" ] }' \
                  http://127.0.0.1:8000/ | jq
[
  {
    "arguments": {
      "filename": "/etc/kea/kea-dhcp4-new.json",
      "size": 3248
    },
    "result": 0,
    "text": "Configuration written to /etc/kea/kea-dhcp4-new.json successful"
  }
]
```

Kea DHCPv6 configuration

Kea DHCPv6 configuration

- the Kea DHCPv6 server is independent from the Kea DHCPv4 server
- both can be started together on the same machine, or on separate machines
- the configuration file for the Kea DHCPv6 server is kea-dhcp6.conf
- the Kea DHCPv6 server can be controlled from the keactrl script or through systemd (on Linux)
- the DHCPv6 configuration can be managed through the Kea Control Agent and Kea Shell

Kea DHCPv6 DUID

- each DHCPv6 server has a unique DHCP-Unique-ID (DUID)
- when re-installing a DHCPv6 server, it might be useful to backup and restore the DUID of the system
- the Kea DHCPv6 DUID is stored in the file `kea-dhcp6-serverid` in the `/var/lib/kea` directory (the path is system/distribution dependent)

Next Webinars

- 14th October - Kea DHCP - Lease allocation, client classification, and option assignment
- 28th October - Kea DHCP - High Availability and Database Backends
- 18th November - Kea DHCP - Monitoring, Logging, and Stork
- 2nd December - Kea DHCP - Migrating to Kea from ISC DHCP

Questions and Answers