
Stork

Release 0.1.0

Dec 02, 2019

Contents

| | | |
|----------|---|-----------|
| 1 | Stork Installation | 3 |
| 2 | Using Stork | 5 |
| 2.1 | Setting up users | 5 |
| 2.2 | Monitoring machines with BIND and Kea | 5 |
| 3 | Stork Backend API | 7 |
| 4 | Developer's Guide | 9 |
| 5 | Manual Pages | 11 |
| 5.1 | stork-server - The central Stork server | 11 |
| 5.1.1 | Synopsis | 11 |
| 5.1.2 | Description | 11 |
| 5.1.3 | Arguments | 11 |
| 5.1.4 | Mailing List and Support | 11 |
| 5.1.5 | History | 11 |
| 5.1.6 | See Also | 12 |
| 5.2 | stork-agent - Stork agent that monitors BIND and Kea services | 12 |
| 5.2.1 | Synopsis | 12 |
| 5.2.2 | Description | 12 |
| 5.2.3 | Configuration | 12 |
| 5.2.4 | Mailing List and Support | 12 |
| 5.2.5 | History | 12 |
| 5.2.6 | See Also | 12 |
| 6 | Indices and tables | 13 |

Stork is a new project proposed by ISC with the aim of delivering BIND and Kea dashboard. It is going to be a spiritual successor of earlier attempts - Kitiwake and Antherius. It is currently in very early stages of planning.

This is the reference guide for Stork version 0.1.0. Links to the most up-to-date version of this document (in PDF, HTML, and plain text formats), along with other documents for Kea, can be found in ISC's [Stork project homepage](#).



Stork Installation

Stork is in its very early stages of development. As such, it is currently only supported on Ubuntu 18.04. It is likely that the code would work on many other systems, but for the time being we want to focus on the core development, rather than portability issues.

There are several dependencies that needs to be installed:

- rake
- Java Runtime Environment
- Docker and Docker Compose

For details, please see Stork wiki <https://gitlab.isc.org/isc-projects/stork/wikis/Development-Environment> . Note the Stork project is in very early stages and its building instructions change frequently. Please refer to the wiki page in case of problems.

For ease of deployment, Stork uses Rake to automate compilation and installation. It currently requires Docker, but soon it will be optional. Technically, you can see how all Stork elements are built and conduct all of those steps manually (without using docker).

The following command will retrieve all required software (go, goswagger, nodejs, Angular dependencies, etc.) to local directory. No root password necessary.

```
# Prepare docker images and start them up
rake docker_up
```

Once the build process finishes, Stork UI will be available at <http://localhost:8080/>. Use any browser to connect.

Note: The installation procedure will create 3 Docker images: *stork_webui*, *stork_server* and *postgres*. The PostgreSQL database schema will be automatically migrated to the latest version required by the Stork server process.

If you run unit-tests, also *stork-ui-pgsql* image will be created. The installation procedure assumes those images are fully under Stork control. If there are existing images, they will be overwritten.

There are several other rake targets. For a complete list of available tasks, use `rake -T`. Also see [wiki](#) for detailed build instructions.

2.1 Setting up users

As of Stork 0.1.0 release, there is no possibility to create new users. This functionality will be added in the next release. Currently, the default administrator's account is created and can be used to sign in to the system via the web UI. Please use the login *admin* and password *admin* to sign in to the system.

2.2 Monitoring machines with BIND and Kea

Todo: Describe how the user can deploy agent, connect new machines, how to monitor them, etc.

CHAPTER 3

Stork Backend API

URL: (stork-url)/version-get

Returns:

```
{  
  "version": "1.2.3"  
}
```


CHAPTER 4

Developer's Guide

Note: We acknowledge that users and developers are two different groups of people, so the documents should eventually be separated. However, since these are still very early days of the project, this section is kept in the Stork ARM for convenience only.

5.1 stork-server - The central Stork server

5.1.1 Synopsis

`stork-server`

5.1.2 Description

The `stork-server` provides the main Stork server capabilities. In every stork deployment, there should be exactly one `stork-server`.

5.1.3 Arguments

Currently `stork-server` takes no arguments.

5.1.4 Mailing List and Support

There is a public mailing list available for the Stork project. **stork-dev** (`stork-dev` at lists.isc.org) is intended for Kea developers, prospective contributors, and other advanced users. The list is available at <https://lists.isc.org>. The community provides best-effort support on both of those lists.

Once stork will become more mature, ISC will be providing professional support for Stork services.

5.1.5 History

The `stork-server` was first coded in November 2019 by Michal Nowikowski and Marcin Siodelski.

5.1.6 See Also

stork-agent (8)

5.2 stork-agent - Stork agent that monitors BIND and Kea services

5.2.1 Synopsis

stork-agent

5.2.2 Description

The `stork-agent` is a small tool that is being run on the systems that are running BIND and Kea services. Stork server connects to the stork agent and uses it to monitor services remotely.

5.2.3 Configuration

Stork agent uses two environment variables to control its behavior:

- `STORK_AGENT_ADDRESS` - if defined, governs which IP address to listen on
- `STORK_AGENT_PORT` - if defined, it controls which port to listen on. The default is 8080.

5.2.4 Mailing List and Support

There is a public mailing list available for the Stork project. **stork-dev** (`stork-dev` at lists.isc.org) is intended for Kea developers, prospective contributors, and other advanced users. The list is available at <https://lists.isc.org>. The community provides best-effort support on both of those lists.

Once stork will become more mature, ISC will be providing professional support for Stork services.

5.2.5 History

The `stork-agent` was first coded in November 2019 by Michal Nowikowski.

5.2.6 See Also

stork-server (8)

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`