Stork

Release 0.2.0

Jan 10, 2020

Contents

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 - Kittiwake and Antherius. It is currently in very early stages of planning.

This is the reference guide for Stork version 0.2.0. Links to the most up-to-date version of this document, along with other documents for Stork, can be found in ISC's Stork project homepage or readthedocs .

CHAPTER 1

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 (when installing using Docker)

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 facilitates installation both using Docker and without Docker (see the following sections).

1.1 Installation using Docker

The following command will retrieve all required software (go, goswagger, nodejs, Angular dependencies, etc.) to your local directory. No root password necessary. Note that Docker installation is optional. Using docker is the easiest installation method. However, if you prefer to install it on your host, see the next section about Native installation.

```
# 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.

1.2 Native Installation

The following steps will install Stork and its dependencies natively, i.e. on the host machine rather than using Docker images.

First, you need to install PostgreSQL. This is OS specific. Please follow up the instructions for your system.

```
$ psql postgres
psql (11.5)
Type "help" for help.

postgres=# CREATE USER stork WITH PASSWORD 'stork';
CREATE ROLE
postgres=# CREATE DATABASE stork;
CREATE DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE stork TO stork;
GRANT
postgres=# \c stork
You are now connected to database "stork" as user "thomson".
stork=# create extension pgcrypto;
CREATE EXTENSION
```

Optional step: if you want to initialize the database on your own, you need to build the migrations and use it to initialize and upgrade the DB to the latest schema. However, this is completely optional as the database migration will be triggered automatically upon the server startup. This is only useful if for some reason you want to set up the database, but don't want to run the server yet. In most cases this step can be skipped.

```
$ rake build_migrations
$ backend/cmd/stork-db-migrate/stork-db-migrate init
$ backend/cmd/stork-db-migrate/stork-db-migrate up
```

Now that you have the database environment set up, the next step is to build all the tools. Note the first command will download some missing dependencies needed and will install it in a local directory. This is done only once and is not needed for future rebuilds. However, it's safe to rerun the command.

```
$ rake build_backend
$ rake build_ui
```

The environment should be ready to run! Open 3 consoles, and run the following 3 commands, one in each console:

```
$ rake run_server
$ rake serve_ui
$ rake run_agent
```

Once all three processes are running, go ahead and connect to http://localhost:4200 with your web browser. See *Using Stork* for initial password information.

CHAPTER 2

Using Stork

This section describes how to use features available in stork. To connect to Stork, use your web browser and connect to port 4200. If Stork is running on your localhost, you can navigate to http://localhost:4200.

2.1 Managing users

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.

To manage users, click on the `Configuration` menu and choose `Users`. You will see a list of existing users. At the very least, there will be user `admin`.

To add new user, click `Create User Account`. A new tab will opened that will let you specify the new account parameters. Some fields have specific restrictions. Username can consist of only letters, numbers and underscore. E-mail field is optional. However, if specified, it must be a well formed e-mail. First and lastname fields are mandatory. Password must only contain letters, digits, @, ., !, +, - and must be at least 8 characters long. Once all requirements are met, the `Save` button will become active and you will be able to add new account.

Note: As of Stork 0.2.0 release, the role-based access control is not implemented yet. Every user is considered a super-admin and has full control over the system.

2.2 Deploying Stork Agent

Stork system uses agents to monitor services. Stork Agent (*STAG* or simply *agent*) is a daemon that is expected to be deployed and run on each machine to be monitored. As of Stork 0.2.0 release there are no automated deployment routines and STAG has to be copied and run manually. This can be done in a variety of ways. Here is one of them.

Assuming you want to monitor services running on machine with IP 192.0.2.1, you can do the following on the Stork server command line:

```
cd <stork-dir>
scp backend/cmd/stork-agent login@192.0.2.1:/path
```

On the machine to be monitored, you need to start the agent. In the basic case, you can simply run it:

```
./stork-agent
```

You can optionally pass `--host=` or set the `STORK_AGENT_ADDRESS` environment variable to specify which address the agent will listen on. You can pass `--port` or set the `STORK_AGENT_PORT` environment variable to specify which TCP port the agent will listen on.

Note: Unless explicitly specified, the agent will listen on all addresses on port 8080. There are no authentication mechanisms implemented in the agent yet. Use with care!

2.3 Registering New Machine

Once the agent is deployed and running on the machine to be monitored, you should instruct Stork server to start monitoring it. You can do so by going to Services menu and choosing Machines. You will be presented with a list of currently registered machines.

To add a new machine, click `Add New Machine`. You need to specify the machine address or hostname and a port. If Stork agent is running in a container, you may specify the container name. This is particularly useful, if you built stork using `rake docker_up` command and the agent is running in a container. In such case, you can use kea-agent as your hostname. If you run agent by using `rake run_agent`, the agent will listen on port 8888.

Once you click Add, the server will attempt to establish gRPC over http/2 connection to the agent. Make sure that any firewalls in between will allow incoming connections to the TCP port specified.

Once a machine is added, number of parameters, such as hostname, address, agent version, number of CPU cores, CPU load, available total memory, current memory utilization, uptime, OS, platform family, platform name, OS version, kernel, virtualization details (if any), host ID and other information will be provided. More information will become available in the future Stork versions.

2.4 Monitoring Machines

To monitor registered machines, go to Services menu and click Machines. A list of currently registered machines will be displayed. Pagination mechanism is available to display larger number of machines.

There is a filtering mechanism that acts as an omnibox. The string typed is searched for an address, agent version, hostname, OS, platform, OS version, kernel version, kernel architecture, virtualization system, host-id fields. The filtering happens once you hit ENTER.

You can inspect the state of a machine by clicking its hostname. A new tab will open with machine details. Multiple tabs can be open at the same time. You can click Refresh state to get updated information.

The machine state can also be refreshed using Action menu. On the machines list, each machine has its own menu. Click on the triple lines button at the right side and choose the Refresh option.

2.5 Deleting Machines

To stop monitoring a machine, you can go to the Machines list, find the machine you want to stop monitoring, click on the triple lines button at the right side and choose Delete. Note this will terminate the connection between Stork server and the agent running on the machine and the server will no longer monitor it. However, the Stork agent process will continue running. If you want to completely shut it down, you need to do so manually, e.g. by connecting to the machine using ssh and stopping the agent there. One way to achieve that is to issue `killall stork-agent` command.

CHAPTER 3

Backend API

Stork agent provides a REST API. The API is generated using [Swagger](https://swagger.io/). The API points are currently documented in the `api/swagger.yaml` file.

Note: In the future Stork releases, the API documentation will be generated automatically.

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.

4.1 Generating Documentation

To generate documentation, simply type `rake doc`. You need to have Sphinx and rtd-theme installed. The generated documentation will be available in the `doc/singlehtml` directory.

4.2 Agent API

The connection between the server and the agents is established using gRPC over http/2. The agent API definition is kept in the `backend/api/agent.proto` file. For debugging purposes, it is possible to connect to the agent using grpcurl tool. For example, you can retrieve a list of currently provided gRPC calls by using this command:

```
$ grpcurl -plaintext -proto backend/api/agent.proto localhost:8888 describe
agentapi.Agent is a service:
service Agent {
  rpc detectServices ( .agentapi.DetectServicesReq ) returns ( .agentapi.
↪DetectServicesRsp );
  rpc getState ( .agentapi.GetStateReq ) returns ( .agentapi.GetStateRsp );
  rpc restartKea ( .agentapi.RestartKeaReq ) returns ( .agentapi.RestartKeaRsp );
}
```

You can also make specific gRPC calls. For example, to get the machine state, the following command can be used:

```
$ grpcurl -plaintext -proto backend/api/agent.proto localhost:8888 agentapi.Agent.
↪getState
{
  "agentVersion": "0.1.0",
  "hostname": "copernicus",
  "cpus": "8",
  "cpusLoad": "1.68 1.46 1.28",
  "memory": "16",
  "usedMemory": "59",
  "uptime": "2",
  "os": "darwin",
  "platform": "darwin",
  "platformFamily": "Standalone Workstation",
  "platformVersion": "10.14.6",
  "kernelVersion": "18.7.0",
  "kernelArch": "x86_64",
  "hostID": "c41337a1-0ec3-3896-a954-a1f85e849d53"
}
```

CHAPTER 5

Manual Pages

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 [**–host**] [**–port**]

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 Arguments

The Stork Agent takes the following arguments:

-h or --help Displays list of available parameters.

--host=hostname Specifies the IP to listen on. Can be controlled with $STORK_AGENT_ADDRESS environment variable. The default value is `::`.

--port=1234 Specifies the TCP port to listen on for connections. The default is 8080. Can be controlled with $STORK_AGENT_PORT environment variable.

5.2.4 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.

5.2.5 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.

5.2.6 History

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

5.2.7 See Also

stork-server (8)

CHAPTER 6

Indices and tables

- genindex
- modindex
- search