

Managing Jethro

Using Service-Related Commands

A Jethro instance includes an entire database – all tables, data, indexes and metadata. One or multiple Jethro hosts can be connected (attached) to the same instance; for example, you can add hosts to handle a larger number of concurrent users (scale-out). Each host can run the following services per attached instance:

- **JethroServer** – A stateless service that receives SQL commands from clients (via ODBC / JDBC / JethroClient), executes these commands and returns the results.
- **JethroMaint** – A background service that runs non-intrusive maintenance operations, such as optimizing indexes and removing unused files. When multiple hosts access the same Jethro instance, only one of them should run this service



If this service is not running over a long period of time you may experience degraded performance, especially after large data loads (the **SHOW TABLES MAINT** command can be used to check this).

Reviewing Common Deployment Options

Jethro instances can be deployed in either of the following deployment methods:

- **Production environment** - Consists of a single Jethro instance, to which multiple hosts are attached. Each of those production hosts has its services automatically running all the time.
- **Development / Test environment** - Consists of one or multiple Jethro hosts, each attached to multiple instances (development / test / training and so on), and not all services are running at the same time. The administrator starts and stops the services on demand (to avoid memory pressure).

Service Start/Stop/Status Commands

The Jethro installation RPM file creates a Linux OS service for starting and stopping Jethro services. It simplifies and automates the startup of Jethro services for production environments, while providing the administrator full control for development/test environments. The service command is run under the *jethro* OS user, so no root privileges is required. It allows selecting a specific instance and/or service to start or stop.

Here's some common examples:

- **Starting all services marked for `auto-start`** (this command also runs automatically during the host startup):

```
service jethro start
Instance demo service JethroMaint started, pid: 24407
Instance demo service JethroServer started, pid: 24506, port: 9112
```

- **Stopping all running services** (also runs during the host shutdown):

```
service jethro stop
```

- **Listing all currently running services:**

```
service jethro status
Instance demo service JethroMaint running, pid: 24407
Instance demo service JethroServer running, pid: 24320 ,port: 9111
Instance demo service JethroServer running, pid: 24506 ,port: 9112
```

- **Starting JethroServer service of a specific instance:**

```
service jethro start demo
```

- **Starting JethroMaint service of a specific instance:**

```
service jethro start demo maint
```

- **Starting both services** of a specific instance:

```
service jethro start demo all
```

Configuring the Automatic Start of Services

The service command automatically monitors any Jethro service that it started (*JethroServer* and *JethroMaint*); upon detecting that a running service was unexpectedly stopped, the *service* command will restart the service.

In addition, the linux service automatically starts Jethro services during boot, based on a text configuration file called *services.ini* that resides under ***lopt/jethro/instances***. This file is automatically populated by the JethroAdmin commands.

For example, the **create-instance** command adds a line that refers to the instance – if the instance being created is the only instance on the host, it will set the value of the autostart parameter to *On*, otherwise it will set autostart to *Off*.

The *services.ini* file has a simple format, where each line controls the auto-start of a specific instance by using the following format:
{instance_name}:{port}:{start_query_service}:{start_maint_service}

For example, with an instance named 'demo', listening at port 9111 (default), with auto-start of both services:
demo:9111:yes:yes

Attaching Additional Jethro Hosts to an Instance

Once a Jethro instance is created, additional Jethro hosts can be configured to access the instance. That enables Jethro to support more concurrent users, as well as to provide high-availability, by using Client-side load balancing where each SQL command is directed to a different Jethro host.

To attach a Jethro host to an existing instance:

1. Prepare a new Jethro host, by following these steps:
2. Set up the Host.
3. [Install Jethro software](#).
4. Verify connectivity.
5. Run the following command to attach the new host to the existing instance, providing the instance name, the instance directory, and the location and size of the local cache directory (similar to the create-instance command):

```
JethroAdmin attach-instance {instance_name}-storage-path={jethro_storage_path} -cache-path={jethro_local_cache_path}-cache-size={size in GB}
```

#For example:

```
JethroAdmin attach-instance demo -storage-path=/user/jethro/instances -cache-path=/mnt/jethro_local_cache -cache-size=80G
```

Listing Locally Attached Instances

After running either the **create-instance** or the **attach-instance** command, the local Jethro host is attached to the HDFS instance. In other words, the HDFS instance is referred from the local Jethro host .

To list all instances that are attached to the local Jethro host, run:

```
JethroAdmin list-instances
```

Setting up Kerberos Instance

Jethro's current support for accessing a Kerberized Hadoop cluster requires the administrator to manually run the *kinit* utility, which is used for obtaining and caching Kerberos ticket-granting tickets for Jethro services. This needs to be done for every Jethro host, as follows:

1. As jethro user, run **kinit** to generate a cache file per each Jethro service/utility (JethroServer, JethroMaint and JethroLoader):

```
kinit -k -t jethro.hadoop.keytab -c jethro.server.cache jethro
kinit -k -t jethro.hadoop.keytab -c jethro.maint.cache jethro
kinit -k -t jethro.hadoop.keytab -c jethro.loader.cache jethro
```



It is recommended to ask for a renewable ticket with long expiration time (based on your site policies), for example adding a **-r999d**

2. In the **\$JETHRO_HOME/instances/instance_name/local-conf.ini** file, remove all comments from the following parameters and set each parameter to the full path of the corresponding cache file:

```
hdfs.kerberos.server.ticket.cache.path
hdfs.kerberos.maint.ticket.cache.path
hdfs.kerberos.loader.ticket.cache.path
```



Because these steps must be repeated when the tickets expire, it is advisable to have a ticket with a long lifetime. In addition, to avoid service disruption, ensure that you periodically renew the Kerberos tickets before they expire, for example by running **kinit -R** from a cron job.

Reviewing Log Files

All log files which contain information with regard to the Jethro services can be found under **/var/log/jethro/{instance_name}**:

1. **jethro_loader.log** – Displays all information regarding JethroLoader's operation
2. **jethroserver.log** – Stores all queries run on Jethro, with details such as the query plans and running time
3. **jethro.log** – A generic log, featuring more detailed log messages
4. **jethro_maint.log** – Displays all information regarding JethroMaint's operations
5. **services.log** – Displays information regarding the service itself, including all issues encountered during service startup or shutdown

Reviewing Configuration Files

The instance configuration files reside under **\$JETHRO_HOME/instances/{instance_name}**:

- **local-conf.ini** – Features the specific configurations for the instance
- **jethrolog.properties** – Controls the log level and information that will be written in the log files specified in topicsection Reviewing Log Files

Removing Jethro

When removing Jethro, you may choose either of the following options:

- **Detaching an Instance** - When a Jethro host is detached from an existing instance, all references to an existing instance are removed from the local Jethro host, without actually deleting the instance.
- **Deleting an Instance** - Removing all of the deleted instance's data, as well as all references to the instance from the local Jethro host.
- In addition, you can remove Jethro software.

Detaching an Instance

To detach a Jethro host from an existing instance, run:

```
JethroAdmin detach-instance {instance-name}
```

#For example:

```
JethroAdmin detach-instance demo
```

The **detach** command also stops all local services that may be running on this instance.



If needed, you can later attach a Jethro host again to the same instance, by running the following command: **JethroAdmin attach-instance**.

Deleting an Instance



If the instance to be deleted is attached to several Jethro hosts, ensure that you detach the instance from all hosts but one, and then run the delete operation from the remaining host.

to delete an instance, run:

```
JethroAdmin delete-instance {instance_name};
```

#For example:

```
JethroAdmin delete-instance demo
```

Removing Jethro Software

To remove *Jethro software*, Switch to **user root** and then run:

```
rpm -e jethro
```



Removing the Jethro software does not delete any instance, because the instances are shared across hosts. If you want to delete an instance, you should do it before you uninstall Jethro software.