# 16477654672_ed59874cce_z.jpg



**Version 1.0**

24 March 2017

Dushyant Min

Akshay Suresh

**Index**

[**Overview:**](#_98x79x5lvhmr) **3**

[Document Purpose](#_fqhvl6g3s9s9) 3

[D30-HectorDA Background:](#_gi4gyyctuusz) 3

[**Installation Instructions:**](#_49j4lxnn5ivv) **4**

[Application Structure:](#_nzro27lsbfz) 4

[Dependencies and Prerequisites:](#_sxi7vy6zqrjh) 4

[Build from source](#_skvbwclxxgvi) 5

[2. Installation](#_umzkd1ny7fxp) 5

[3. Validate and Manage Hector](#_3eluvihfmod) 5

[4. Log File Explained:](#_sp0dbhalssls) 6

# 

# 

# Overview:

## Document Purpose

This document is created with the purpose :

* To give a brief background of D30-HectorDA application.
* How to install the application
* The configuration options available in the application.
* How to perform common activities with the application

## D30-HectorDA Background:

D30-HectorDA application is a middleware application that exposes APIs to clients, in the background it can connect to different types of data sources and answers queries and execute insert requests for the client. This application serves as an abstraction layer to make accessing different types of data easy without having to implement the specific semantics of the target data source.

The present client interfaces supported are GRPC using Protobuf 3 over HTTP and a REST API interface.

# Installation Instructions:

## Application Structure:

This is a Golang based application that uses Supervisord to make sure that the application is running at all times. Supervisord basically starts the HectorDA application as a child process and restarts it when it sees that the application fails.

The application is packaged nicely in an RPM so installation is as simple as rpm -ivh on the target machine. Because this is an application that is under active development and keeps getting updated, we have given instructions to pull the latest code and run the package script that generates the RPM. The section one can be run on a different machine than where you want to do the actual installation. First step needs an internet connection so you could do it on a different server and then just take the RPM and install it on the actual server where the installation needs to be done.

The configuration of the application is a toml file that can be found in /etc/hector/config.toml after the installation is done.

After the install is done all you need to do on the target machine is start Supervisord.

## 

## Dependencies and Prerequisites:

* Python 2.7 - ([Suggested Install Instructions](https://www.digitalocean.com/community/tutorials/how-to-set-up-python-2-7-6-and-3-3-3-on-centos-6-4))



* Go 1.7 ( [Suggested Install Instructions](https://tecadmin.net/install-go-on-centos/#) )
  + Setup GOROOT and GOPATH
* Supervisor 3.1.3 ( Suggested Install Instructions )
* Git
* Rpm-build

## Build from source



This step pulls the latest code and generates an RPM. This RPM can then be used directly on the machine or added to your local yum repository and then you can install the hector-da application direct by using

* Clone the latest code

**git clone https://github.com/dminGod/D30-HectorDA.git**

* Run the following command:

**cd D30-HectorDA**

**scripts/package**

This will give an RPM in the ~/rpmbuild/RPMS/noarch

## 2. Installation



* On the target machine run the following command

**rpm -ivh <rpm-file>**

* Go to /etc/hector/config.toml
  + Make sure that the proper hosts for cassandra and presto are added
* Run the following

**service supervisord start**

## 3. Validate and Manage Hector

The following command will show you the state of the hector application:

**> supervisordctl**

**hector RUNNING pid 444, uptime 0:15:42**

To start and stop the service the following commands can be sent to supervisord

**supervisordctl stop hector**

**supervisordctl start hector**

## 4. Log File Explained:

# This is the configuration of Hector Server

# Location : /etc/hector/config.toml

# DO NOT DELETE THIS FILE

version = "0.0.1"

# Hector Server Config

[hector]

connectionType = "tcp"

version = "0.0.1"

port = "5001" **// Port where Hector will start and listen on**

log = "INFO" **// Log level, can be DEBUG, INFO, ERROR**

logDirectory = "conf-example/" **// Where hector will store its logs**

startServersOfType = ["grpc"] **// grpc and http can be passed here --**

**http is currently in experimental mode**

requestMetrics = false **// Controls what metrics get stored on ELK**

queryMetrics = false **// Controls what metrics get stored on ELK**

# Cassandra Config

[cassandra]

host = ["127.0.0.1"] **// Location of the cassandra server**

port = "9042" **// Port for cassandra**

# **Presto Details**

[presto]

connectionURL = "presto://127.0.0.1:8080/cassandra/all\_trade"