```
(defn queue->dsdb [{:keys [queue-name queue dsdb]}]
  ;(timbre/debug "Checking for new items in queue...")
  (when-some [task (dq/take! queue queue-name 10 nil)]
    (do
      (timbre/debug "Putting data into datascript")
      (d/transact! dsdb [(line->record @task)])
      (dq/complete! task))))

(def config
  {::jdbc/connection       {:connection-uri "jdbc:h2:mem:mem_only"}
   ::jdbc/init             {:conn (ig/ref ::jdbc/connection)}
   ::datascript/connection w/schema
   ::hawk/watch            {:groups [{:paths   ["example"]
                                      :handler #'file-handler}]
                            :queue  (ig/ref ::durable/queues)
                            :conn   (ig/ref ::jdbc/connection)}
   ::durable/queues        {:delete-on-halt? true
                            :directory       "/tmp"}
   ::scheduling/job        {:job        #'queue->dsdb
                            :queue-name :my-queue
                            :schedule   {:in [5 :seconds] :every :second}
                            :queue      (ig/ref ::durable/queues)
                            :dsdb       (ig/ref ::datascript/connection)}
   ::web/server            {:host     "0.0.0.0"
                            :port     3000
                            :sql-conn (ig/ref ::jdbc/connection)
                            :conn     (ig/ref ::datascript/connection)
                            :handler  #'handler}})

(defonce sys (create config))
```

```clojure
(defn queue->dsdb [{:keys [queue-name queue dsdb]}]
  ;(timbre/debug "Checking for new items in queue...")
  (when-some [task (dq/take! queue queue-name 10 nil)]
    (do
      (timbre/debug "Putting data into datascript")
      (d/transact! dsdb [(line->record @task)])
      (dq/complete! task))))

(def config
  {::jdbc/connection       {:connection-uri "jdbc:h2:mem:mem_only"}
   ::jdbc/init             {:conn (ig/ref ::jdbc/connection)}
   ::datascript/connection w/schema
   ::hawk/watch            {:groups [{:paths   ["example"]
                                      :handler #'file-handler}]
                            :queue  (ig/ref ::durable/queues)
                            :conn   (ig/ref ::jdbc/connection)}
   ::durable/queues        {:delete-on-halt? true
                            :directory       "/tmp"}
   ::scheduling/job        {:job        #'queue->dsdb
                            :queue-name :my-queue
                            :schedule   {:in [5 :seconds] :every :second}
                            :queue      (ig/ref ::durable/queues)
                            :dsdb       (ig/ref ::datascript/connection)}
   ::web/server            {:host     "0.0.0.0"
                            :port     3000
                            :sql-conn (ig/ref ::jdbc/connection)
                            :conn     (ig/ref ::datascript/connection)
                            :handler  #'handler}})

(defonce sys (create config))
```