

### Coordinator

#### Overview

The Druid Coordinator process is primarily responsible for segment management and distribution. More specifically, the Druid Coordinator process communicates to Historical processes to load or drop segments based on configurations. The Druid Coordinator is responsible for loading new segments, dropping outdated segments, managing segment replication, and balancing segment load.

### Router

#### Overview

The Apache Druid (incubating) Router process can be used to route queries to different Broker processes. By default, the broker routes queries based on how Rules are set up. For example, if 1 month of recent data is loaded into a hot cluster, queries that fall within the recent month can be routed to a dedicated set of brokers. Queries outside this range are routed to another set of brokers. This set up provides query isolation such that queries for more important data are not impacted by queries for less important data.

### Overlord

#### Overview

The Overlord process is responsible for accepting tasks, coordinating task distribution, creating locks around tasks, and returning statuses to callers. Overlord can be configured to run in one of two modes - local or remote (local being default). In local mode Overlord is also responsible for creating Peons for executing tasks. When running the Overlord in local mode, all MiddleManager and Peon configurations must be provided as well. Local mode is typically used for simple workflows. In remote mode, the Overlord and MiddleManager are run in separate processes and you can run each on a different server. This mode is recommended if you intend to use the indexing service as the single endpoint for all Druid indexing.

### Historical

### Broker

#### Overview

The Broker is the process to route queries to if you want to run a distributed cluster. It understands the metadata published to ZooKeeper about what segments exist on what processes and routes queries such that they hit the right processes. This process also merges the result sets from all of the individual processes together. On start up, Historical processes announce themselves and the segments they are serving in Zookeeper.

### Middle Manager

#### Overview

The MiddleManager process is a worker process that executes submitted tasks. Middle Managers forward tasks to Peons that run in separate JVMs. The reason we have separate JVMs for tasks is for resource and log isolation. Each Peon is capable of running only one task at a time, however, a MiddleManager may have multiple Peons.



By [kinger1999](#)

[cheatography.com/kinger1999/](https://cheatography.com/kinger1999/)

Not published yet.

Last updated 19th June, 2019.

Page 1 of 1.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>