

sinksFactory.BuildAll(),对传入参数进行判断,然后调用this.Build()进行创建,这里只需要注意即使没有配置metric,也会进行metricSink的创建。NewDataSinkManager,为每个sink创建协程,等待数据的到来并最终将数据导入到对应的后端数据库

NewKubeletProvider创建数据源,创建kubernetes master及kubelet client相关的配置,创建kubeClient及kubeletClient,创建协程进行watch,便于后面调用nodeLister.List()列出所有的nodes。

数据传递到sinkManager, ma nager依次将数据传递给sinkH olders, sinkHolder再调用sin k导出数据。 cluster\_aggregator:聚合ns的监控指标到cluster下 namespace\_aggregator:聚合pod监控指标到namespace下 namespace based enricher:给所有的监控数据加上namespace id

node\_aggregator:聚合pod监控指标到node下

node\_autoscaling\_enricher:聚合node资源使用情况pod\_aggregator:聚合container监控指标到pod下

pod\_based\_enricher:简单说就是给container加上pod的信息

rate calculator: 计算时间, 使用率等指标。

NewManager 根据sourceManager, dataProcessors, sinkManager对象,再创建manager。

housekeep 根据cAdvisor就是支持时间区间来获取metrics值。 ScrapeMetrics,通过nodeLister获取k8s所有的nodes,这样便能知道 所有的kubelet信息,然后创建对应数量的协程从各个kubelet中获取对应 的cAdvisor监控信息,进行处理后再返回。

ExportData, 创建协程, 然后将之前获取数据放入dataBatchChannel