Autoscale Elasticsearch reindexing on K8S

Team

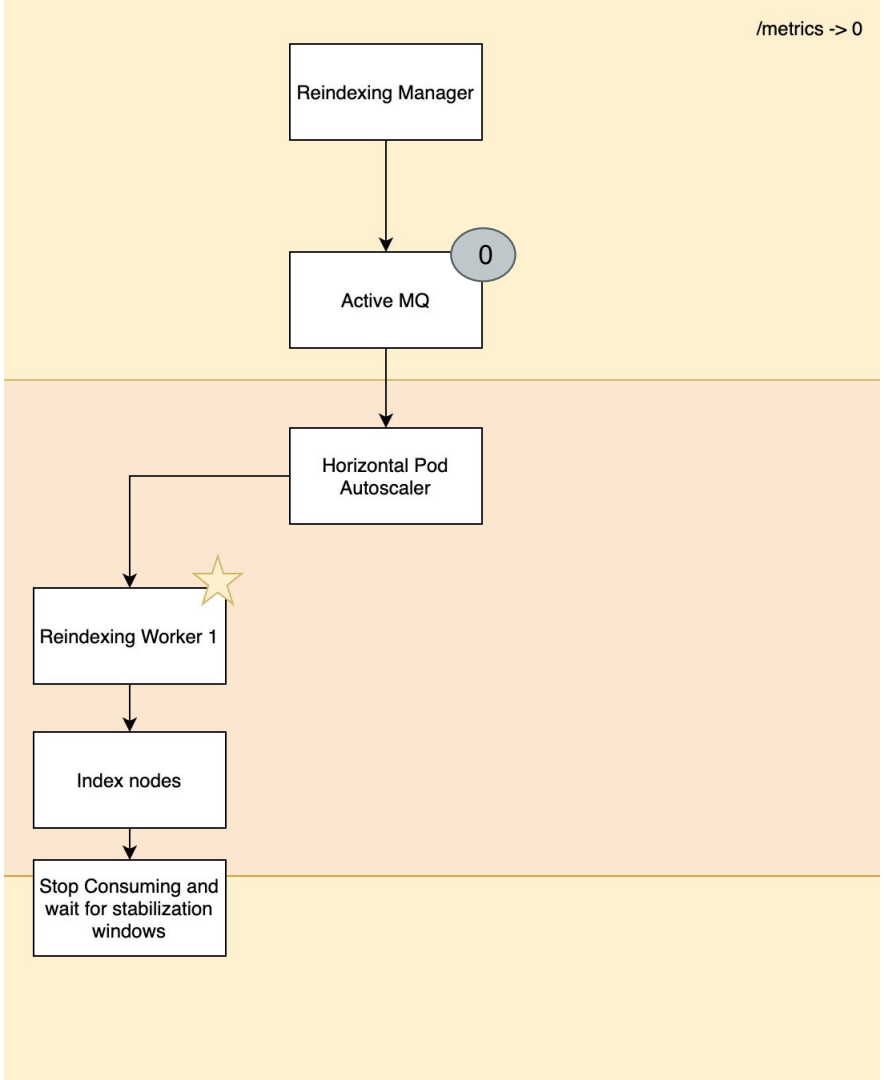
Davide Cerbo

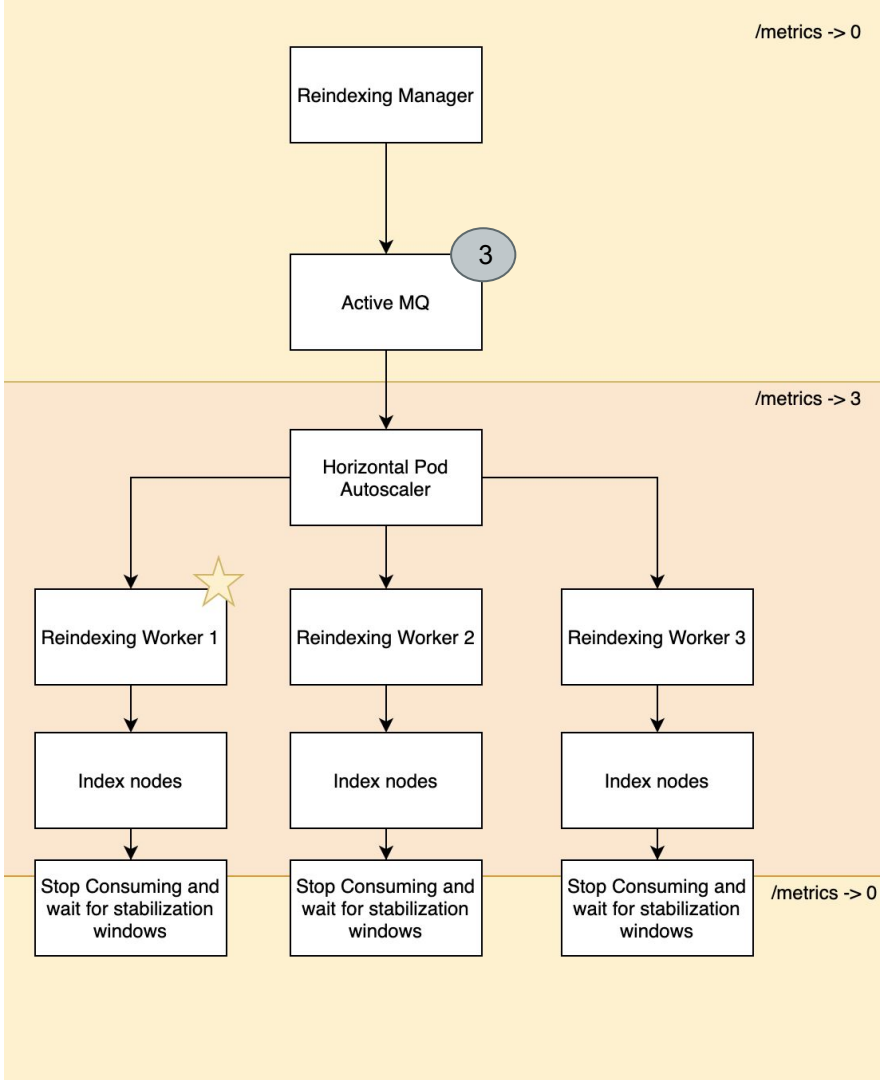
Roberto Franchini

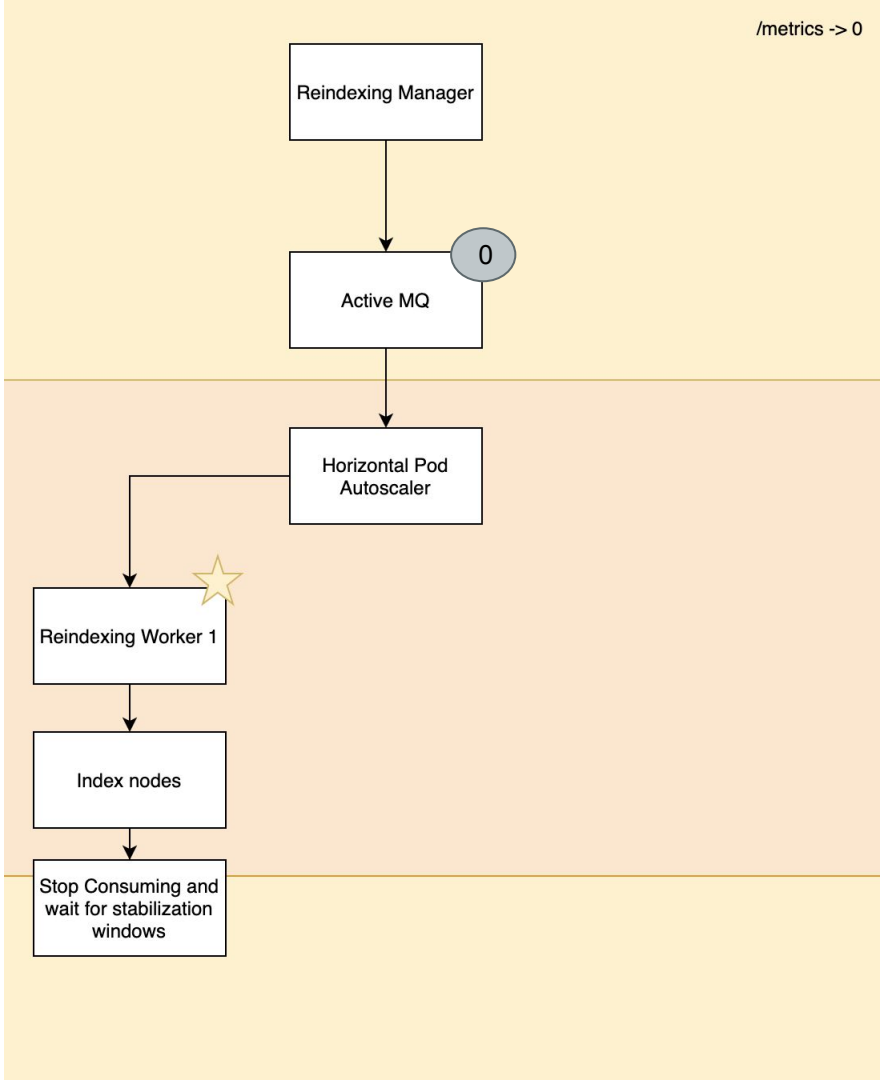
Georgios Evangelopoulos

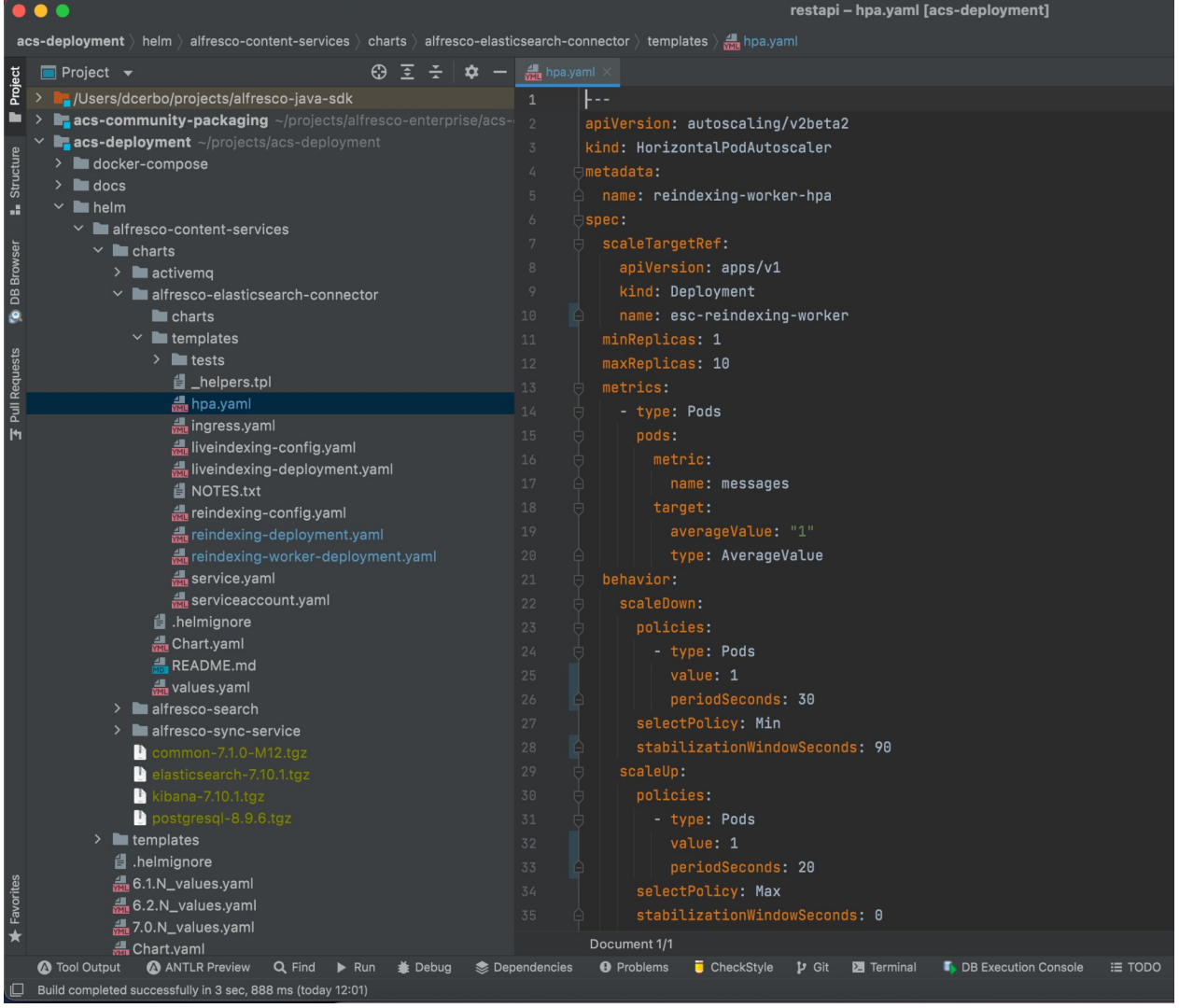
Keerat Lalia

Scaling Elasticsearch reindexing based on partitions number.


/metrics -> 0
Reindexing Manager
0
Active MQ
Horizontal Pod Autoscaler
Reindexing Worker 1
Index nodes
Stop Consuming and wait for stabilization windows


/metrics -> 0
Reindexing Manager
3
Active MQ
/metrics -> 3
Horizontal Pod Autoscaler
Reindexing Worker 1
Reindexing Worker 2
Reindexing Worker 3
Index nodes
Index nodes
Index nodes
Stop Consuming and wait for stabilization windows
Stop Consuming and wait for stabilization windows
Stop Consuming and wait for stabilization windows
/metrics -> 0


/metrics -> 0
Reindexing Manager
0
Active MQ
Horizontal Pod Autoscaler
Reindexing Worker 1
Index nodes
Stop Consuming and wait for stabilization windows


restapi – hpa.yaml [acs-deployment]
acs-deployment 〉 helm 〉 alfresco-content-services 〉 charts 〉 alfresco-elasticsearch-connector 〉 templates 〉 hpa.yaml
Project
/Users/dcerbo/projects/alfresco-java-sdk
acs-community-packaging ~/projects/alfresco-enterprise/acs-
acs-deployment ~/projects/acs-deployment
docker-compose
docs
helm
alfresco-content-services
charts
activemq
alfresco-elasticsearch-connector
charts
templates
tests
_helpers.tpl
hpa.yaml
ingress.yaml
liveindexing-config.yaml
liveindexing-deployment.yaml
NOTES.txt
reindexing-config.yaml
reindexing-deployment.yaml
reindexing-worker-deployment.yaml
service.yaml
serviceaccount.yaml
.helmignore
Chart.yaml
README.md
values.yaml
alfresco-search
alfresco-sync-service
common-7.1.0-M12.tgz
elasticsearch-7.10.1.tgz
kibana-7.10.1.tgz
postgresql-8.9.6.tgz
templates
.helmignore
6.1.N_values.yaml
6.2.N_values.yaml
7.0.N_values.yaml
Chart.yaml
hpa.yaml
1 ---
2 apiVersion: autoscaling/v2beta2
3 kind: HorizontalPodAutoscaler
4 metadata:
5 name: reindexing-worker-hpa
6 spec:
7 scaleTargetRef:
8 apiVersion: apps/v1
9 kind: Deployment
10 name: esc-reindexing-worker
11 minReplicas: 1
12 maxReplicas: 10
13 metrics:
14 - type: Pods
15 pods:
16 metric:
17 name: messages
18 target:
19 averageValue: "1"
20 type: AverageValue
21 behavior:
22 scaleDown:
23 policies:
24 - type: Pods
25 value: 1
26 periodSeconds: 30
27 selectPolicy: Min
28 stabilizationWindowSeconds: 90
29 scaleUp:
30 policies:
31 - type: Pods
32 value: 1
33 periodSeconds: 20
34 selectPolicy: Max
35 stabilizationWindowSeconds: 0
Document 1/1
Tool Output
ANTLR Preview
Find
Run
Debug
Dependencies
Problems
CheckStyle
Git
Terminal
DB Execution Console
TODO
Build completed successfully in 3 sec, 888 ms (today 12:01)

```java
public class ReindexingController {

  private final ReindexingService reindexingService;
  private final QueueMetricsService queueMetricsService;

  public ReindexingController(ReindexingService reindexingService, QueueMetricsService queueMetricsService) {
    this.reindexingService = reindexingService;
    this.queueMetricsService = queueMetricsService;
  }


  @GetMapping("/reindex")
  public HttpStatus reindex() {
    try {
      reindexingService.reindex();
      return HttpStatus.OK;
    } catch (Exception e) {
      return HttpStatus.INTERNAL_SERVER_ERROR;
    }
  }

  @ResponseBody
  @RequestMapping(value="/metrics", produces="text/plain")
  public String metrics() {
    return "# HELP messages Number of messages in the queueService\n"
        + "# TYPE messages gauge\n"
        + "messages " + queueMetricsService.fetchPartitions();
  }

}
```


BOOM!
DEMO

Do you remember GraalVM?

(The previous hackathon)

Where we can apply the same concepts?

- Live indexing (Metadata, Path, Content events)
- Every event-driven service

Thanks
https://github.com/learnk8s/spring-boot-k8s-hpa