Instructions & Tasks

You work as a research librarian who is currently studying the works of Shakespeare; specifically, *Romeo and Juliet*. You have a 3-node Elasticsearch cluster and the complete works of Shakespeare, which you use for your literary analysis. Currently, the complete works of Shakespeare are indexed to a single index called shakespeare, but, since you are currently focused on the play *Romeo and Juliet*, you would prefer to copy this play to its own index.

To accomplish this, you will first need to create a new index called romeo\_and\_juliet with the same field mappings as the shakespeare index. Since your 3-node Elasticsearch cluster only has 2 data nodes, you want to create the romeo\_and\_juliet index with 2 primary shards and 1 replica shard. Once the romeo\_and\_juliet index has been created, you will need to use the \_reindex API to copy all documents with play\_name of "Romeo and Juliet" to the romeo\_and\_juliet index.

In addition to copying the data for the play *Romeo and Juliet* to its own index, you also want to modify the data in-flight during the reindexing process. Specifically, you want to take the contents of the field text\_entry and store each whitespace-delimited word in an array called word\_array. Additionally, you want to add a word\_count field that is equal to the number of words in the word\_array field. Lastly, because the index will only contain data for the play *Romeo and Juliet*, we can remove the play\_name field. All of this can be accomplished with an ingest node pipeline using the split, script, and remove processors.

Your 3-node Elasticsearch cluster is already up and running on the master node, along with a Kibana instance. You can perform the instructions above with either Kibana's console tool or curl on the command line. If you want to use Kibana's console tool for this activity, you will need to perform Task 1. Otherwise, skip Task 1 and continue to Task 2.

help

check\_box\_outline\_blank

**OPTIONAL: Establish a remote tunnel to Kibana on the master node.**

**NOTE: This task only needs to be performed if you're using the Kibana UI to interact with Elasticsearch.**

Set up a Remote Tunnel

1. Open a new terminal window and SSH to the master node as cloud\_user but with port forwarding

ssh cloud\_user@your\_public\_ip -L 5601:localhost:5601

Open the Kibana Console Tool

1. In your local web browser, go to http://localhost:5601.
2. In Kibana, navigate to **Dev Tools** via the side navigation bar.
3. Select the **Console** tool (it should be the default tool that loads).

help

check\_box\_outline\_blank

**Create the `romeo\_and\_juliet` index.**

Option 1: Kibana Console Tool

Create the Index

1. To create the romeo\_and\_juliet index, use the **Console** to send the following request to Elasticsearch:
2. PUT romeo\_and\_juliet
3. {
4. "mappings": {
5. "doc": {
6. "properties": {
7. "line\_id": {
8. "type": "integer"
9. },
10. "line\_number": {
11. "type": "text",
12. "fields": {
13. "keyword": {
14. "type": "keyword",
15. "ignore\_above": 256
16. }
17. }
18. },
19. "play\_name": {
20. "type": "keyword"
21. },
22. "speaker": {
23. "type": "keyword"
24. },
25. "speech\_number": {
26. "type": "integer"
27. },
28. "text\_entry": {
29. "type": "text",
30. "fields": {
31. "keyword": {
32. "type": "keyword",
33. "ignore\_above": 256
34. }
35. }
36. },
37. "type": {
38. "type": "text",
39. "fields": {
40. "keyword": {
41. "type": "keyword",
42. "ignore\_above": 256
43. }
44. }
45. }
46. }
47. }
48. },
49. "settings": {
50. "number\_of\_shards": 2,
51. "number\_of\_replicas": 1
52. }

}

Option 2: Command-line curl

Create the Index

1. To create the romeo\_and\_juliet index, execute the following from the command line of one of the nodes:
2. curl -XPUT "http://localhost:9200/romeo\_and\_juliet" -H 'Content-Type: application/json' -d'
3. {
4. "mappings": {
5. "doc": {
6. "properties": {
7. "line\_id": {
8. "type": "integer"
9. },
10. "line\_number": {
11. "type": "text",
12. "fields": {
13. "keyword": {
14. "type": "keyword",
15. "ignore\_above": 256
16. }
17. }
18. },
19. "play\_name": {
20. "type": "keyword"
21. },
22. "speaker": {
23. "type": "keyword"
24. },
25. "speech\_number": {
26. "type": "integer"
27. },
28. "text\_entry": {
29. "type": "text",
30. "fields": {
31. "keyword": {
32. "type": "keyword",
33. "ignore\_above": 256
34. }
35. }
36. },
37. "type": {
38. "type": "text",
39. "fields": {
40. "keyword": {
41. "type": "keyword",
42. "ignore\_above": 256
43. }
44. }
45. }
46. }
47. }
48. },
49. "settings": {
50. "number\_of\_shards": 2,
51. "number\_of\_replicas": 1
52. }

}'

help

check\_box\_outline\_blank

**Create the `shakespeare-tokenizer` ingest node pipeline.**

Option 1: Kibana Console Tool

Create the Ingest Pipeline

1. To create the ingest pipeline, use the **Console** to send the following request to Elasticsearch:
2. PUT \_ingest/pipeline/shakespeare-tokenizer
3. {
4. "description": "Tokenizes the text\_entry field into an array. Adds a word\_count field. Removes the play\_name field.",
5. "processors": [
6. {
7. "split": {
8. "field": "text\_entry",
9. "separator": "\\s+",
10. "target\_field": "word\_array"
11. }
12. },
13. {
14. "script": {
15. "lang": "painless",
16. "source": "ctx.word\_count = ctx.word\_array.length"
17. }
18. },
19. {
20. "remove": {
21. "field": "play\_name"
22. }
23. }
24. ]

}

Option 2: Command-line curl

Create the Ingest Pipeline

1. To create the ingest pipeline, execute the following from one of the node's command line:
2. curl -XPUT "http://localhost:9200/\_ingest/pipeline/shakespeare-tokenizer" -H 'Content-Type: application/json' -d'
3. {
4. "description": "Tokenizes the text\_entry field into an array. Adds a word\_count field. Removes the play\_name field.",
5. "processors": [
6. {
7. "split": {
8. "field": "text\_entry",
9. "separator": "\\s+",
10. "target\_field": "word\_array"
11. }
12. },
13. {
14. "script": {
15. "lang": "painless",
16. "source": "ctx.word\_count = ctx.word\_array.length"
17. }
18. },
19. {
20. "remove": {
21. "field": "play\_name"
22. }
23. }
24. ]

}'

help

check\_box\_outline\_blank

**Reindex the play "Romeo and Juliet".**

Option 1: Kibana Console Tool

Reindex the Data

1. To reindex the data, use the **Console** to send the following request to Elasticsearch:
2. POST \_reindex
3. {
4. "source": {
5. "index": "shakespeare",
6. "query": {
7. "match": {
8. "play\_name": "Romeo and Juliet"
9. }
10. }
11. },
12. "dest": {
13. "index": "romeo\_and\_juliet",
14. "pipeline": "shakespeare-tokenizer"
15. }

}

Option 2: Command-line curl

Reindex the Data

1. To reindex the data, execute the following from one of the node's command line:
2. curl -XPOST "http://localhost:9200/\_reindex" -H 'Content-Type: application/json' -d'
3. {
4. "source": {
5. "index": "shakespeare",
6. "query": {
7. "match": {
8. "play\_name": "Romeo and Juliet"
9. }
10. }
11. },
12. "dest": {
13. "index": "romeo\_and\_juliet",
14. "pipeline": "shakespeare-tokenizer"
15. }

}'