SWE 265P Reverse Engineering and Modeling

Homework 1: System features and UML class diagram

Team name: Skateboard 2 wheelchair

Team members: Zeyu Huang, Yue Zhang, Tianlun Li

Project: ElasticSearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

Feature 1. IP Filter

IP filtering is an Elasticsearch security feature, we can restrict connections by applying IP filtering to application clients, node clients, or transport clients, in addition to other nodes that are attempting to join the cluster. If a node's IP address is on the blacklist, the Elasticsearch security features allow the connection to Elasticsearch but it is dropped immediately and no requests are processed.

To find out how the ip filter feature was implemented, we first searched the keyword ‘ip filter’ in the project and started from that class **IPFilter**. This class has method **accept()** to determine whether the input ip is allowed by the **SecurityIpFilterRule** we defined. In **SecurityNetty4HttpServerTransport** and **SecurityNetty4ServerTransport**, we find IpFilter is one of the fields of the two classes and the **doStart** method calls the **setBoundTransportAddress** and **setBoundHttpTransportAddress** methods of IpFilter. We believe it’s for applying different IP filtering for the transport and HTTP protocols. And finally in Security.java, the two classes we found above were passed in as parameters in getTransports() method.

Folder	File	Method	Relevant ?	Relevant how?	Confidence (1-5)
Org.elasticsearch.xpack.sec	IPFilter.java	accept()	Yes	This method determines if	5

urity.transport.filter				an address can be accepted	
Org.elasticsearch.xpack.security.transport.filter	SecurityIpFilterRule.java		Yes	It's a decorator for netty's IpFilterRule class, it defines the rules for ip filtering	5
org.elasticsearch.xpack.security.transport.filter.netty4	SecurityNetty4HttpServerTransport.java	Constructor；doStart()	Yes	The doStart() method called setBoundHttpTransportAddress method of ipFilter	4
org.elasticsearch.xpack.security.transport.filter.netty4	SecurityNetty4ServerTransport.java	Constructor；doStart()	Yes	The doStart() method called setBoundTransportAddress method of ipFilter.	4
org.elasticsearch.xpack.security	Security.java	getTransports()	Yes	SecurityNetty4ServerTransport and SecurityNetty4HttpServerTransport.java are provided as parameters	4

Feature 2. Rank Evaluation

The Ranking Evaluation API lets you quickly evaluate the quality of search results for a typical query set. This can be useful either while developing new search queries, incremental improvements of the query templates of an existing system, or as a basic monitoring tool to detect changes in the search quality of a system in production.

To understand how this feature implemented, we started from searching keywords ‘evaluation’ and found the class named **EvaluationMetric** which provides a way to compute the quality metric for a result list returned by some search hits and a list of rated documents. Through looking for the usage of the class, we found **RankEvalSpec.** This class groups the queries to evaluate, including their document ratings, and the evaluation metric including its parameters. In this class, an instance of the **EvaluationMetric** is created and used as the quality metric. After several times of searching, we finally found that the **RankEvalAction** the **RankEvalPlugin** classes. These two classes shows that the feature was implemented as action type and was integrated as a plugin which can be easily used by customers and other developers.

Folder	File	Method	Relevant?	Relevant how?	Confidence
org.elasticsearch.index.rankeval	EvaluationMetric.java	combine()	Y	comblie() access EvalQueryQuality:: metricScore	5
org.elasticsearch.index.rankeval	ExpectedReciprocalRank.java		Y	implements EvaluationMetric interface	5
org.elasticsearch.index.rankeval	EvalQueryQuality.java	toXContent()	Y	Calls EvaluationMetric.filterUnratedDocuments(ratedHits)	5
org.elasticsearch.index.rankeval	TransportRankEvalAction	doExecute()	Y	EvaluationMetric metric =	2

				evaluationSpecification.getMetric();	
org.elasticsearch.index.rankeval	RankEvalPlugin	getActions	Y	return Arrays.asList(new ActionHandler<>(RankEvalAction.INSTANCE, TransportRankEvalAction.class));	2
org.elasticsearch.index.rankeval	RankEvalRequestIT	nodePlugins	Y	return Arrays.asList(RankEvalPlugin.class);	1
org.elasticsearch.index.rankeval	RankEvalSpec	RankEvalSpec	Y	This class groups the queries to evaluate, including their document ratings, and the evaluation metric including its parameters.	5
	RestRankEvalAction	parseRankEvalRequest	Y	RankEvalSpec spec = RankEvalSpec.parse(parser); rankEvalRequest.setRankEvalSpec(spec);	
	RankEvalPlugin	getRestHandlers			5

org.elasticsearch.xpack.core	XPackPlugin	getRestHandlers			3