For the participants, the project started in a London apartment where Shay Banon was busy looking for a job while his wife was studying cooking at the Le Cordon Bleu cooking school. In his spare time, he began to write search engines to help his wife manage more and more rich recipes. The goal is simple and the direction is clear. With the rapid increase in usage of the later version, the software began to have its own community and attracted great attention, especially from Steven Schuurman, Uri Bones and Simon Willnauer. The four eventually formed a search company. As the real ElasticSearch participants, the four started the gradual formation of the project. The early division of the project directly and independently took some time. After a period of time, the confusion of the technology stack brought many difficult problems. Participants then began to adjust their development methods, unify the numbering and naming methods, and launch Elastic search and Kibana services on AWS through Elastic Cloud (formerly Found). Through the division of labor of the four participants, the structure of the project is classified, and the next step is to design under the separated software architecture.

The stakeholders of Elasticsearch are categorized according to the stakeholder types described in Rozanski & Woods (2012), including three additional types: competitors, investors, and marketers. The categories have been placed in alphabetical order.

Shay banon, because his wife is studying cooking at the cooking school, thinks about using search engine to manage recipes with the continuous enrichment of recipes. He first built the elasticsearch framework, then released it to the public open source, and created the elasticsearch IRC channel. With more and more users, he quickly attracted the participation of Steven Schuurman, URI boness and Simon willnauer. Before and after the founding of elasticsearch Inc., two other open-source projects are also developing by leaps and bounds.

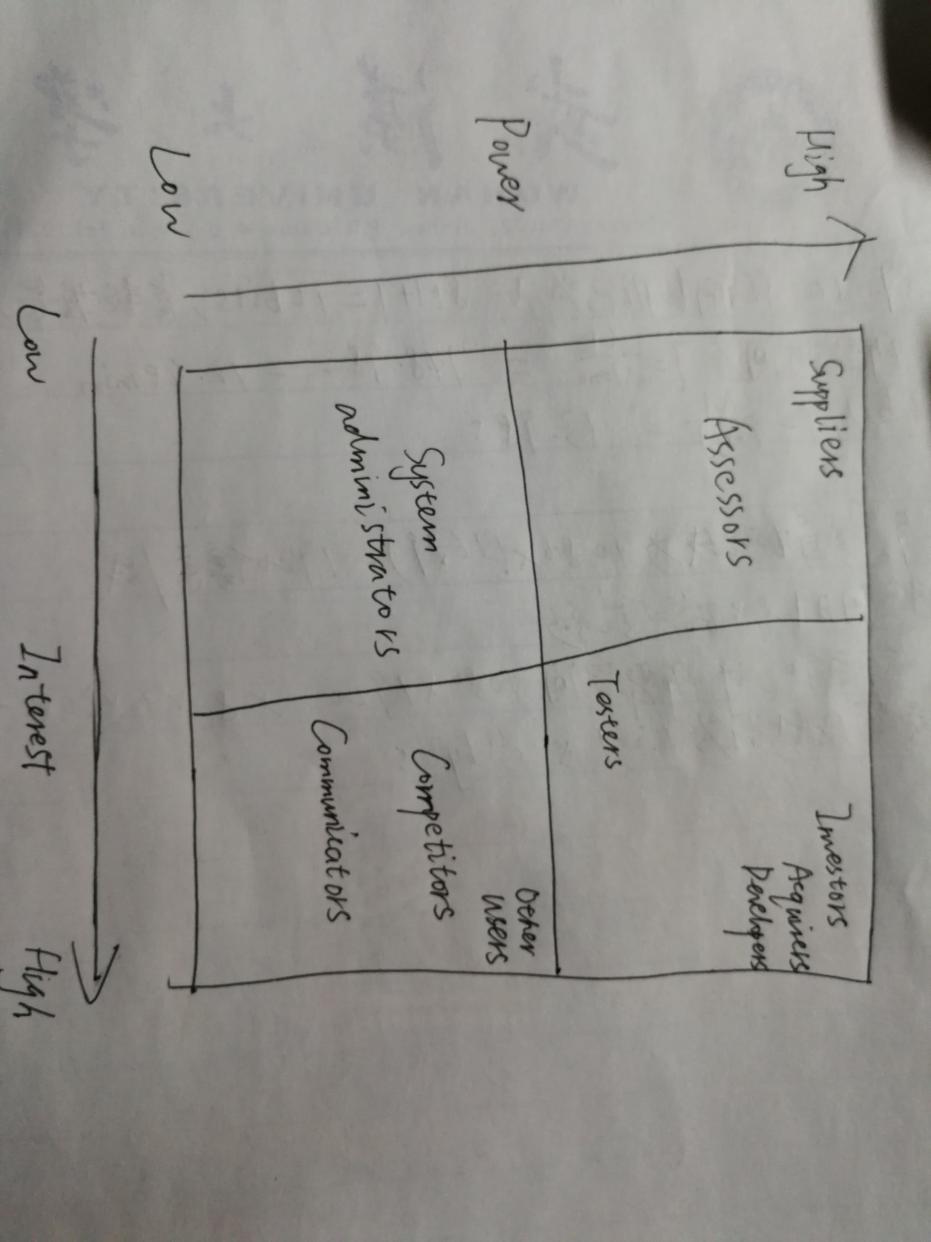
Jordan Sissel was developing logstash, an open-source pluggable data collection tool that can send log files to the "repository" selected by users. For Rashid Khan, he was developing an open source UI called kibana.

Shay, Jordan and Rashid have known each other for a while and know each other's products quite well, so they finally decided to work together for common development. Elk stack officially appeared, namely: elasticsearch, logstash and kibana stack.

Soon two commercial plug-ins were launched: marvel for monitoring and shield for protection.

The cluster construction of ES is very simple, it does not need to rely on the third-party coordination management components, and the cluster management function is realized internally. Es cluster consists of one or more elasticsearch nodes. Each node can join the cluster by configuring the same cluster.name. The default value is "elasticsearch". Ensure that different cluster names are used in different environments, otherwise nodes will eventually join the wrong cluster. An elastic search service startup instance is a node. The node uses node.name to set the node name. If not, the node is assigned a random universal unique identifier as the name at startup. This part contributes a lot to the analysis and propagation of the system. This part is mainly completed by URI bones.

The grid is shown below:



Each of the stakeholder categories mentioned above has been included in the grid.

Each primary partition has one or more copies. When the primary partition is abnormal, a copy can provide data query and other operations. The primary partition and the corresponding replica partition are not on the same node, so the maximum number of replica partitions is n-1 (where n is the number of nodes). The new, index, and delete requests for documents are all write operations, which must be completed on the primary partition before they can be copied to the relevant replica partition. In order to improve the ability of writing, ES writes concurrently. At the same time, in order to solve the problem of data conflict in the process of concurrent writing, ES controls by optimistic locking. Each document has a version number. When the document is modified, the version number increases. Once all the copies are successfully written, the coordination node will report success to the coordination node, and the coordination node will report success to the client. This part is mainly completed by Simon Willnauer.