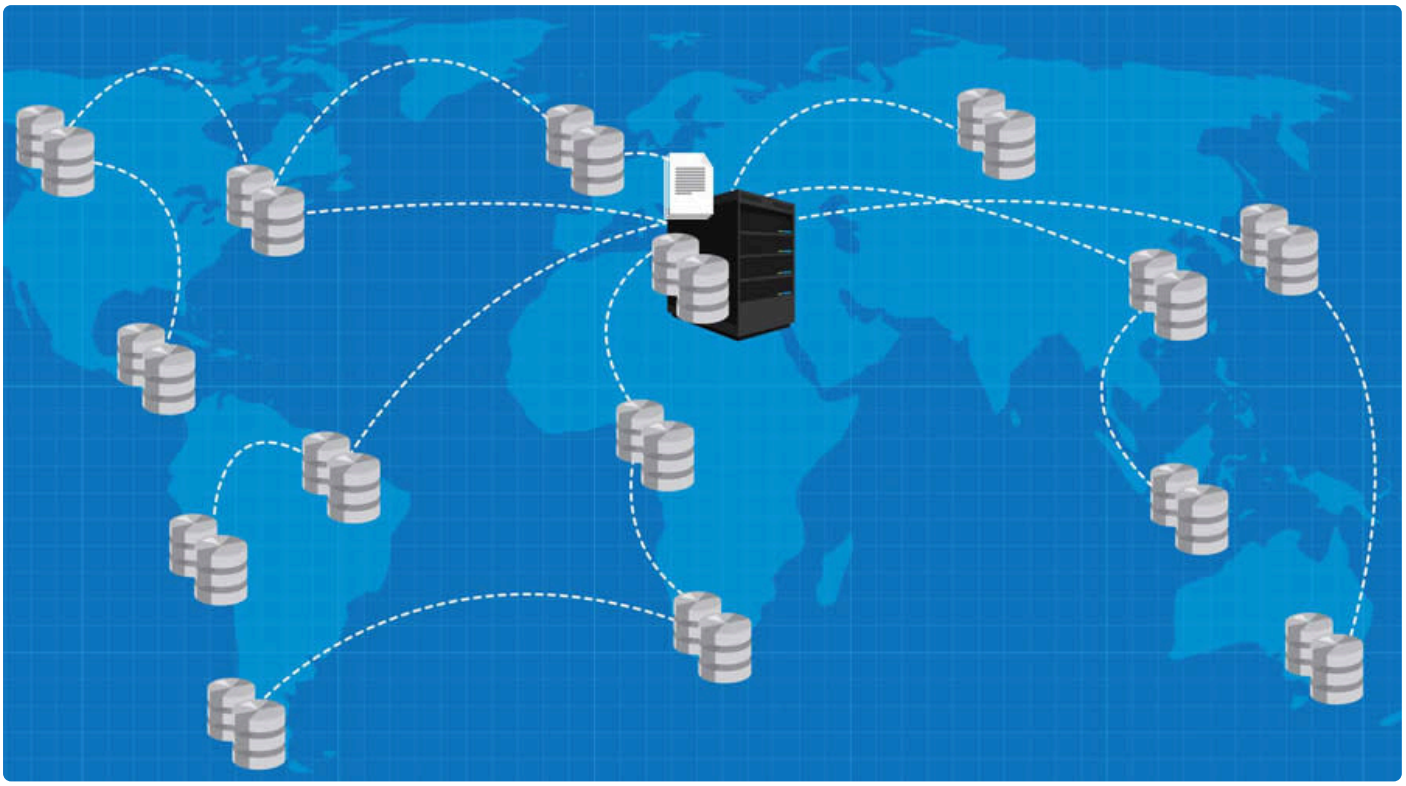
Security for distributed systems

Cliff Staley · 2018-02-20



Security in a non-distributed system seems to be a piece of cake when compared to problems we deal with nowadays. Systems were running on a single machine, and its components were separate operating system (OS) processes. Communication between these processes was protected out of the box by security mechanisms implemented in the OS.

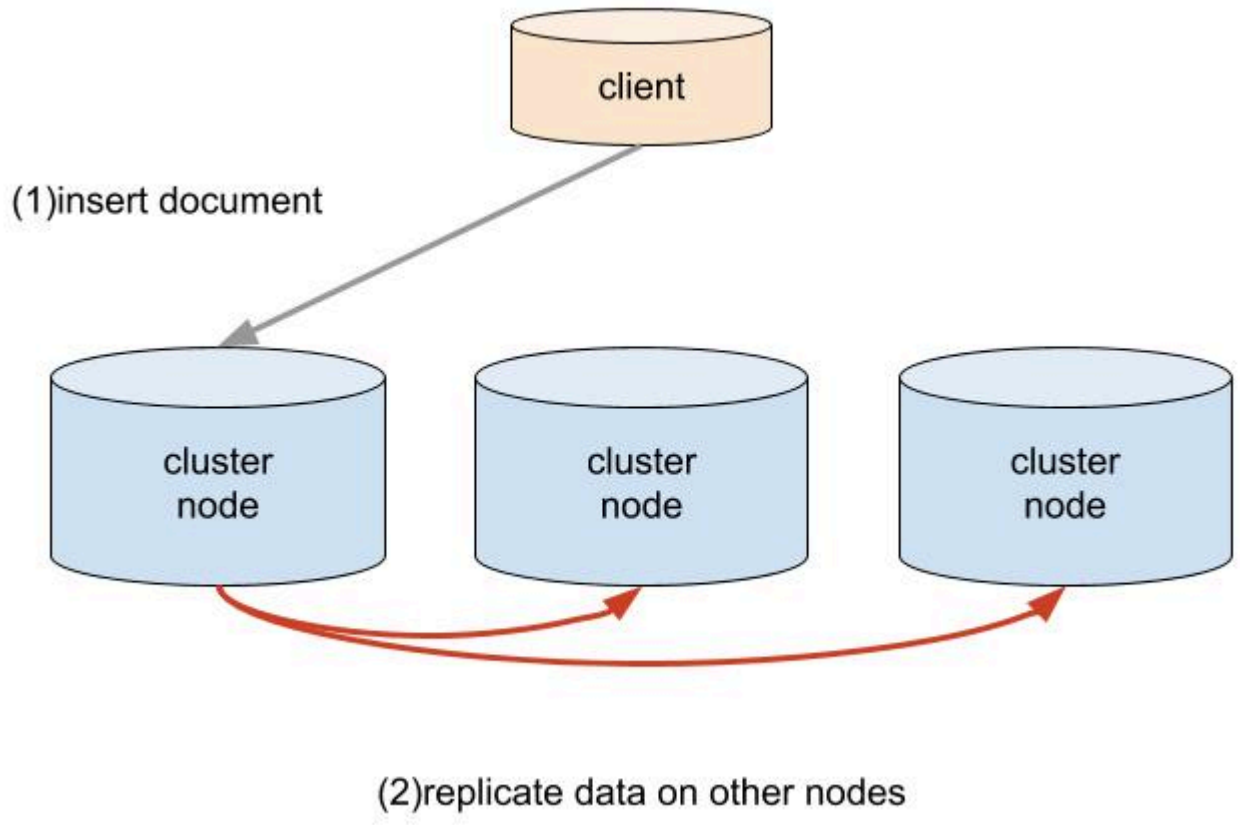
But non-distributed systems are not able to scale well. In the current era of distributed systems, clusters of data span across multiple servers. This provides performance and scalability advantages but also introduces extra difficulties when it comes to security that did not exist earlier. In this article, we focus on some critical features of distributed systems and see how do they affect security.

Distributed Systems principles and paradigms

High availability

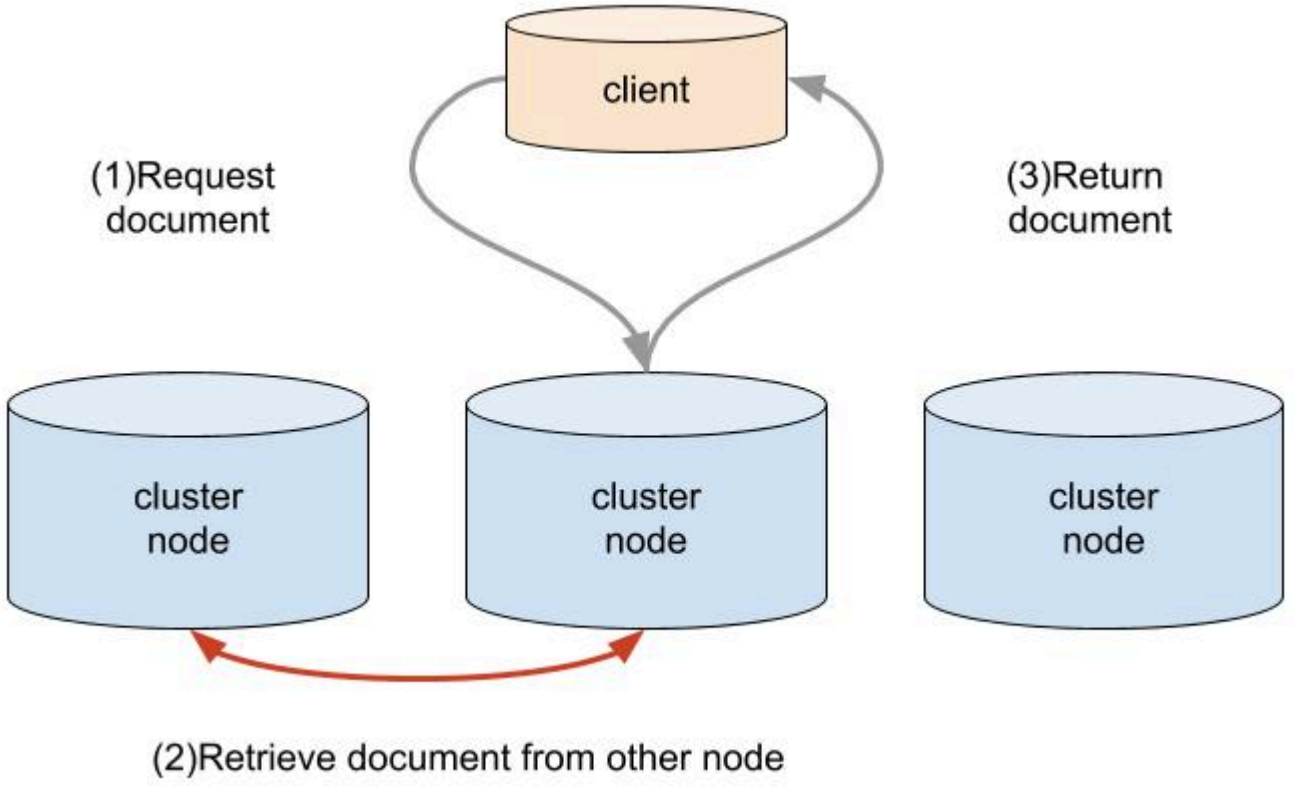
The first one is high availability. Which means we want a distributed system to run even if some machines go down. Mainly, we do not want to lose data, so data needs to be replicated across cluster nodes. Most frequently it is the cluster that takes care and maintains replicas of your data: A client sends data to a single node, and the system replicates it on other nodes. The same applies to Elasticsearch (ES). When a client sends a document to a single node, ES sends it to other nodes containing the same index partition.

This impacts security a lot. Not only should we take care of securing the connection from a client to the cluster. Inter-node communication between cluster nodes has to be secured as well.



Avoid bottlenecks

In order to scale there should not be any component that becomes a bottleneck of the whole system. To achieve that, clients communicate with multiple nodes as this distributes server workload and network traffic. The same is true for Elasticsearch. When a client issues a search request, the receiving node may have all data to fulfill it. But since your data is distributed amongst all nodes in the cluster, it is more likely that it has to talk to other nodes as well. This, again, requires inter-node communication. The red arrow on the bottom shows this inter-node communication, which needs to be secured as it also transfers sensitive data.



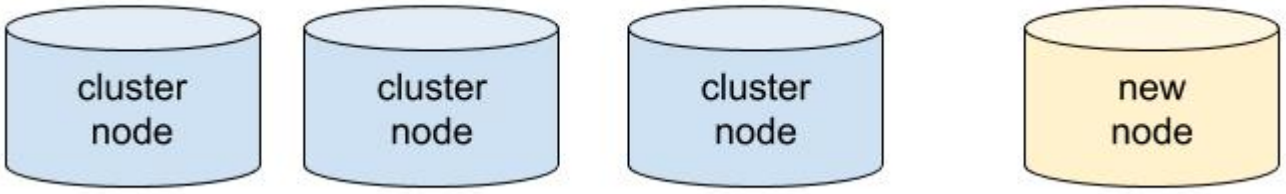
Scale dynamically

Another critical feature of a distributed system is the ability to scale dynamically. It means you need to be able to scale your cluster when the amount of data or the amount of traffic grows. This is an area where Elasticsearch shines: Just throw some additional nodes at your cluster, and let Elastic do the rest. While scaling becomes extremely easy, it also introduces a potential security risk: Because new nodes will be used for data replication, we need to make sure that nodes joining the cluster are identified and trusted reliably.

Although not related to Elasticsearch directly, the June 2017 data breach at OneLogin[1] revealed that an attacker had used this pattern to steal data and compromise their platform:

> Our review has shown that a threat actor obtained access to a set of AWS keys and used them to access the AWS API from an intermediate host with another, smaller service provider in the US. [...] Through the AWS API, the actor created several instances in our infrastructure to do reconnaissance.

(Source: https://www.onelogin.com/blog/may-31-2017-security-incident)



Apart from Search Guard, no non-commercial solution prevents unauthenticated nodes from joining an Elasticsearch cluster.

How to secure a distributed system?

Transport Layer Security (TLS) between cluster nodes****

In the previous sections, we have shown that:

- Data transferred between cluster nodes need to be encrypted.
- Data has to be transferred between authenticated cluster nodes only.

Internal company networks alone surely do not guarantee that. An article[1] of Google research engineers compares a perimeter security to a medieval castle: a fortress surrounded by a moat and a heavily guarded entry to it. Everything is fine until someone passes a gate.

This strategy worked well in the past, but its fundamental assumptions are not valid anymore as the employees may work from home (outside the castle) and organizations allow external workers to enter the castle (freelancers and contractors).

> Perimeter-based security, NAC's, firewalls and VPNs are ineffective against malicious insiders and targeted attacks.

(Forrester Report, https://www.cyxtera.com/forrester-report-no-more-chewy-centers)

This means that **Transport Layer Security between cluster nodes is a must have.**

Secure cluster nodes directly

Another idea may be to set up a proxy in front of Elasticsearch cluster. This looks good at first glance but is still similar to medieval castles' security. Once an attacker gains access to a proxy, we are in trouble.

There is a wrong assumption in that conception that says "a proxy understands the logic of operations it passes". It may be true when a client inserts new documents, searches an index or retrieves some documents by their IDs. The idea gets tricky when dealing with operations like: multi search, multi get, delete by query and much more. We explained that in the article "Don't rely on proxies for Elasticsearch security".

Proxies do not understand the operations they process. Based on that, they cannot decide if an operation should be allowed or not. Only cluster nodes are capable of making such decisions because:

- They know metadata,
- They genuinely understand the operation that client requests.

This leads us to another rule: **Security must be enforced directly on cluster nodes.**

Distributed Systems and GDPR

EU GDPR (European Union General Data Protection Regulation) does not consider distributed and non-distributed systems separately. It calls encryption as a security requirement and obliges organizations to conduct risk assessments:

- What happens when network traffic between Elasticsearch nodes is sniffed or altered?
- What happens when an attacker gains access to Elasticsearch nodes?

GDPR requires organisations to answer such questions and mitigate those risks.

References

1. https://www.onelogin.com/blog/may-31-2017-security-incident
2. http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/43231.pdf

Image: shutterstock / Aa Amie

This article was published on the Search Guard blog: https://search-guard.com/blog/security-for-distributed-systems/