UiPath

ElasticSearch Minimum Deployment

3 Node Architecture

Revision History

Date	Version	Author	Description
1st August 2016	2016.1	Andrei Bogdan	Edited document
18th August 2016	2016.1	Andrei Bogdan	Add details for ElasticSearch network & discovery zen module
14th August 2017	2017.1	Ovidiu Bestea	New ElasticSearch Architecture

Summary

Overview

Elasticsearch is a near real-time search platform. There is a slight delay between the moment a document is indexed until it becomes searchable. Elasticsearch is a distributed search engine that supports multiple nodes, providing scalability and high availability.

Basic ElasticSearch Concepts

Cluster

A cluster is a collection of one or more server nodes that collectively hold your data and provide indexing and searching capabilities across all nodes. A cluster is identified by a unique name, which is "elasticsearch" by default. The name is important, because a node can only become part of a cluster if its name is identical to those of all the other nodes.

Node

A node is a single server that is part of your cluster, stores your data, and contributes to the cluster's indexing and searching capabilities. Same as a cluster, a node is identified by a name, which by default is a random Marvel character name assigned at the startup time. The default node name can be edited for administrative purposes, to enable the identification of the correspondence between the servers in your network and the nodes in your Elasticsearch cluster.

Index

An index is a collection of documents with relatively similar characteristics, identified by a name in lowercase letters. When the contained documents are added to the index, searched, updated, or deleted, the name is used to refer to the corresponding index.

Document

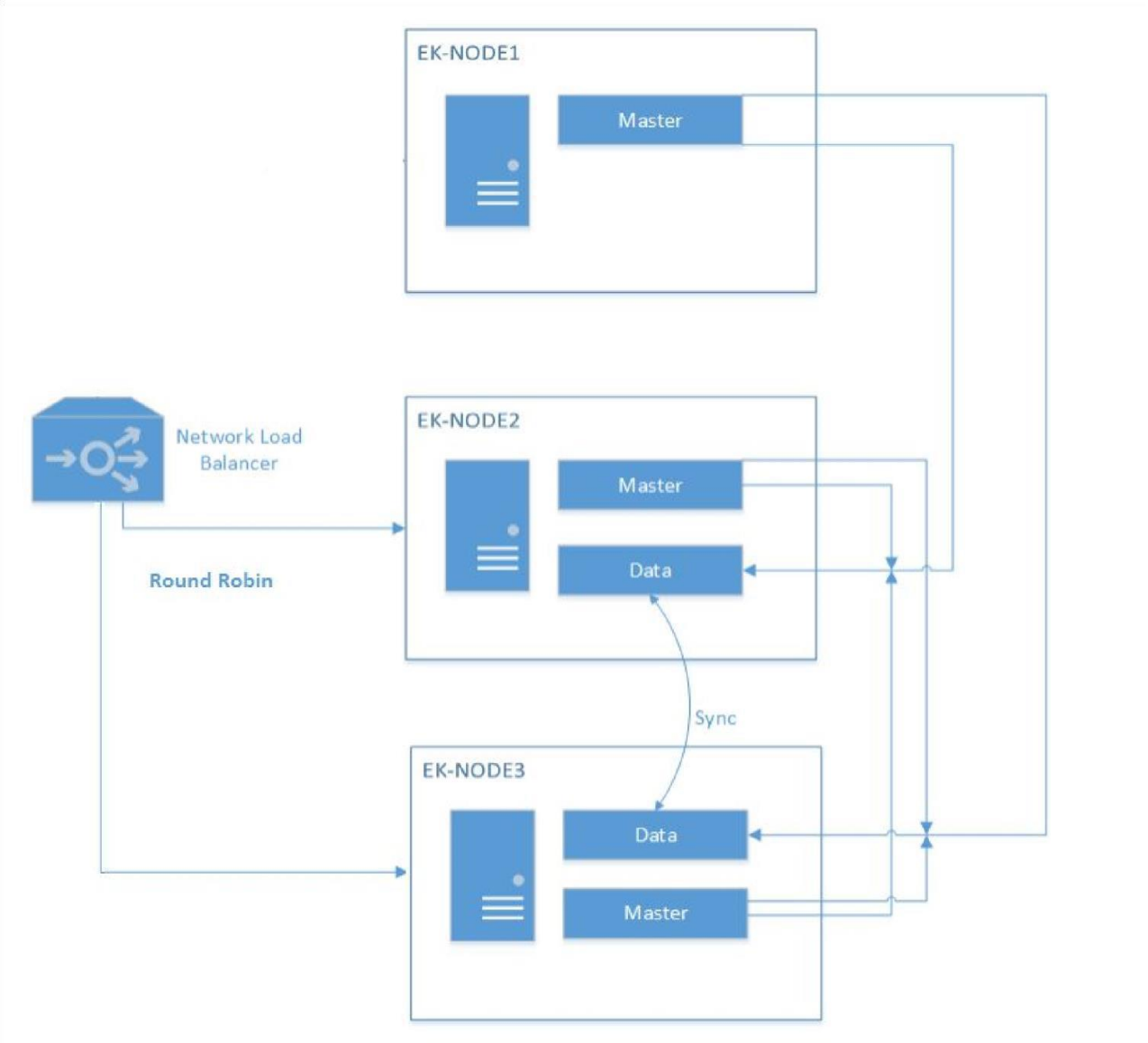
A document is a basic indexable information unit expressed in JSON (JavaScript Object Notation), which is an ubiquitous internet data interchange format.
For more information, visit the ElasticSearch documentation webpage:
https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

1. Cluster configuration

Elasticsearch nodes can be configured with one of the following roles:

- **Master:** The node that maintains the global cluster state, adjusts it according to the current needs, and handles the addition and removal of nodes. There can only be a single active master node in a cluster.
- **Data:** The node that holds data and executes data-related operations (indexing and searching) on the contained shards.
- **Client:** The node that handles index and search requests, forwarding the former to the appropriate primary shard, and the latter, to all the relevant shards. Afterwards, the Client Node aggregates the results.

By default, each node can have the Master, Data, or Client role, as well as two different roles – both Master and Data, for instance. In large and highly-loaded clusters, it is very important to clearly divide the roles of the nodes, by assigning a single role to each one of them. Most often when dealing with such clusters, there are at least three Master Nodes, multiple Data Nodes, and a few Client-only Nodes.



The importance of data integrity is often underestimated. To ensure fault tolerance and availability, a cluster containing a minimum of three nodes/servers is required. If a cluster contains three or more servers, the failure of a single Elasticsearch node does not cause it to fail entirely.

When using a NLB (Network Load Balancer), traffic should be directed to both the Master and the Data Nodes in a round-robin balancing algorithm.

2. Hardware requirements

Each node should meet the following minimum/recommended configuration. Please note that, ideally, the Master and the Data nodes should be allocated identical hardware resources. The requirements of the Master-only node are much lower.

Master Node - recommended configuration:

- RAM: 4 GB
- CPU: 2 CPU(s) 1.8+ GHz
- Disk: 50 GB HDD
- Network: 1 GbE

Master & Data Nodes - minimum configuration:

- RAM: 8 GB
- CPU: 2 CPU(s) 1.8+ GHz
- Disk: 250 GB SSD/15k RPM HDD
- Network: 1 GbE

Master & Data Nodes - recommended configuration:

- RAM: 16/32 GB
- CPU: 4/8 CPU(s) 2.0+ GHz
- Disk: 500 GB/1 TB SSD/15k RPM HDD
- Network: 1 GbE

3. Prerequisites

Let's assume there is a NLB called ek-nlb with the IP address of 10.10.10.201, and there are 3 nodes. Find the settings of each node below.

Node 1:
Name: ek-node1 IP: 10.10.10.245
ElasticSearch Roles: Master

Node 2:
Name: ek-node2 IP: 10.10.10.244
ElasticSearch Roles: Master & Data

Node 3:
Name: ek-node3 IP: 10.10.10.243
ElasticSearch Roles: Master & Data

Cluster name: orchestrator-develop
Ek-node2 and ek-node3 have to be added to the load balancer, directs the entire traffic to both of them according to a round-robin algorithm. The same version of Elasticsearch and Kibana needs to be installed on both nodes, which share the same configuration. It is also recommended to install the Elasticsearch head plugin on each node, to be able to monitor the cluster status.

To perform the installation, perform the following steps:

- Open the command prompt;
- Go to the Elasticsearch bin directory (ex. C:\Elk\elasticsearch\bin);
- Run the following command: plugin install mobz/elasticsearch-head

4. ElasticSearch configuration

Find the Elasticsearch and Kibana configuration for each node below. To see more details about the network settings, please access this link:
https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.html

To see more details about the discovery zen settings, please access this link:
https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery-zen.html

In the case of the three nodes below, we used the computer name as the value of the "node.name" parameter; any descriptive name can be provided instead. Keep in mind that in the examples below, ek-node1, ek-node2, and ek-node3 are the real computer names in our environment.

Ek-node1 (node1) configuration
elasticsearch.yml
cluster.name: orchestrator-develop
node.name: ek-node1
node.data: false
node.master: true
network.host: 10.10.10.245
network.bind_host: 0.0.0.0
http.port: 9200
discovery.zen.ping.unicast.hosts: ["ek-node1", "ek-node2", "ek-node3"]
discovery.zen.minimum_master_nodes: 2 discovery.zen.no_master_block: all
gateway.expected_data_nodes: 2
kibana.yml
server.port: 5601
server.host: "0.0.0.0"
elasticsearch.url: http://10.10.10.245:9200

Ek-node2 (node2) configuration
elasticsearch.yml
cluster.name: orchestrator-develop node.name: ek-node2
node.data: true
node.master: true
network.host: 10.10.10.244
network.bind_host: 0.0.0.0
http.port: 9200
discovery.zen.ping.unicast.hosts: ["ek-node1", "ek-node2", "ek-node3"]
discovery.zen.minimum_master_nodes: 2 discovery.zen.no_master_block: all
gateway.expected_data_nodes: 2
kibana.yml
server.port: 5601
server.host: "0.0.0.0"
elasticsearch.url: "http://10.10.10.244:9200"

Ek-node3 (node 3) configuration
elasticsearch.yml
cluster.name: orchestrator-develop node.name: ek-node3
node.data: true
node.master: true
network.host: 10.10.10.243

network.bind_host: 0.0.0.0
http.port: 9200
discovery.zen.ping.unicast.hosts: ["ek-node1", "ek-node2", "ek-node3"]
discovery.zen.minimum_master_nodes: 2 discovery.zen.no_master_block: all
gateway.expected_data_nodes: 2
kibana.yml
server.port: 5601
server.host: "0.0.0.0"
elasticsearch.url: "http://10.10.10.243:9200"