Elasticsearch

Elasticsearch یک موتور جست و جو متن باز (Open Source) و بلادرنگ (real-time) برای تحلیل متون، جستجو و ذخیره‌سازی انواع داده های نیمه ساختاریافته به صورت توزیع پذیر است .

موتورهای جستجو

موتورهای جستجو (search engines) ابزارهایی هستند که به طور خاص برای نیازمندی‌های پیچیده‌ی کاربران در حوزه‌ی جستجو طراحی شده‌اند.

در این بین مواردی هستند که به آنها online search engine یا موتور جستجوی آنلاین گفته می‌شود که به طور خاص روی محتوای وب و فعالیت‌های آنلاین تمرکز داشته و داده ی ورودی آنها محتواهایی است که افراد مختلف در بستر شبکه ی اینترنت منتشر می‌کنند.

روش کار موتورهای جستجو

فرایند کار موتورهای جستجو شامل ۳ مرحله‌ی اساسی است:

۱) جمع‌آوری داده:

این مرحله در موتورهای جستجوی آنلاین به مرحله‌ی Crawling معروف است. معنای لغوی crawl خزیدن است و به مرحله‌ای گفته می‌شود که طی آن موتورهای جستجو با استفاده از ابزارها یا ربات‌هایی که در اختیار دارند، شروع به خواندن اطلاعات صفحه‌های وبسایت‌های مختلف کرده و داده‌های منتشر شده در آنها را جمع‌آوری می‌کنند. برخی موتورهای جستجو که به صورت آنلاین کار نمی‌کنند، در این مرحله به صورت دستی (manually) داده‌های ورودی آنها تامین می‌شود.

۲) شاخص‌گذاری :

این مرحله در اصطلاح (indexing) شاخص‌گذاری نامیده می‌شود.

کلمات کلیدی داده‌ها مورد بررسی قرار گرفته و برای ثبت شدن در پایگاه داده دسته‌بندی و استخراج می‌شوند.

۳) رتبه‌بندی:

در زمان جستجوی یک عبارت، مجموعه‌ای از الگوریتم‌ها استفاده می‌شود تا امتیازی برای میزان ارتباط هر یک از داده‌ها با عبارت جستجو شده به دست آید. در نهایت بر مبنای امتیاز به دست آمده نتایج مرتب شده و مرتبط‌ترین آنها به کاربر نمایش داده می‌شود.

Elasticsearch یک موتور جست و جو متن باز (Open Source) و بلادرنگ (real-time) برای تحلیل متون، جستجو و ذخیره‌سازی انواع داده های نیمه ساختاریافته به صورت توزیع پذیر (distributable) است.

متن باز بودن (open source) : Elasticsearch با زبان برنامه نویسی جاوا توسعه داده شده است و سورس کد آن در سایت github.com قابل دسترس است.

بلادرنگ بودن : Elasticsearch پس از ذخیره داده ها در کمترین فاصله آنها را برای جستجو آماده می‌کند.
همان طور که در بخش قبل توضیح داده شد، دومین مرحله‌ی کار موتورهای جستجو، indexing یا شاخص‌گذاری کلمات کلیدی آنهاست.

Elasticsearch به طور پیش‌فرض یک ثانیه پس از ثبت داده ها شروع به index کردن آنها می‌کند.

Elasticsearch یک موتور جست و جو متن باز (Open Source) و بلادرنگ (real-time) برای تحلیل متون، جستجو و ذخیره‌سازی انواع داده های نیمه ساختاریافته به صورت توزیع پذیر (distributable) است.

تحلیل و جستجوی : منظور از تحلیل کردن داده ها، تبدیل آنها به فرمتی بهینه برای عملیات full text search است. پس از آن Elasticsearch می‌تواند به سرعت تمامی داده های مرتبط با یک عبارت مورد نظر را شناسایی و امتیازدهی کند.

پشتیبانی از داده های نیمه ساختاریافته : منظور از داده‌ی نیمه ساختاریافته این است که تمامی داده ها لزومی ندارد ساختار کاملا یکسان داشته باشند. برای مثال فرض کنید داده های logبازدید را داشته باشیم و برخی از این logها دارای یک فیلد به نام event_idباشند و برخی دیگر آن را نداشته باشند اما در عین حال همگی آنها شامل فیلدهای مشترک source، created_at و typeباشند. در داده های ساختاریافته (structured) که به طور معمول در پایگاه داده ی رابطه ای از آنها استفاده می‌شود، تمامی داده ها باید ساختار کاملا یکسانی داشته باشند و برعکس در داده های بدون‌ساختار (unstructured) هیچ شمای مشخصی در هیچ یک از فیلدهای داده‌ای آن وجود ندارد مانند داده های ویدیویی.

Elasticsearch یک موتور جست و جو متن باز (Open Source) و بلادرنگ (real-time) برای تحلیل متون، جستجو و ذخیره‌سازی انواع داده های نیمه ساختاریافته به صورت توزیع پذیر (distributable) است.

توزیع پذیری : Elasticsearch به گونه‌ای طراحی شده است که بتوان منابع مورد استفاده‌ی آن را با کم ترین دردسر گسترش داد تا در صورت نیاز بتواند از حجم بالاتر داده ها و پردازش‌های سنگین پشتیبانی کند. طراحی Elasticsearch بر اساس معماری cluster یا خوشه است. در این معماری تعدادی سرور که در یک شبکه به یکدیگر متصل هستند به عنوان یک سیستم واحد عمل کرده اند و ضمن ارتباط با یکدیگر درخواست‌های دریافت شده را پاسخ می‌دهند

ساختار ذخیره و بازیابی داده ها در Elasticsearch

Elasticsearchداده ها را در فرمت JSONمی‌شناسد. در Elasticsearchبه جای ساختار کلاسیک جدول ها و ستون‌ها در پایگاه داده ی رابطه ای، مفاهیمی به نام index و document معرفی می‌شود.

می‌توان index را به عنوان جدول و document را ردیفی از جدول تصور کرد.

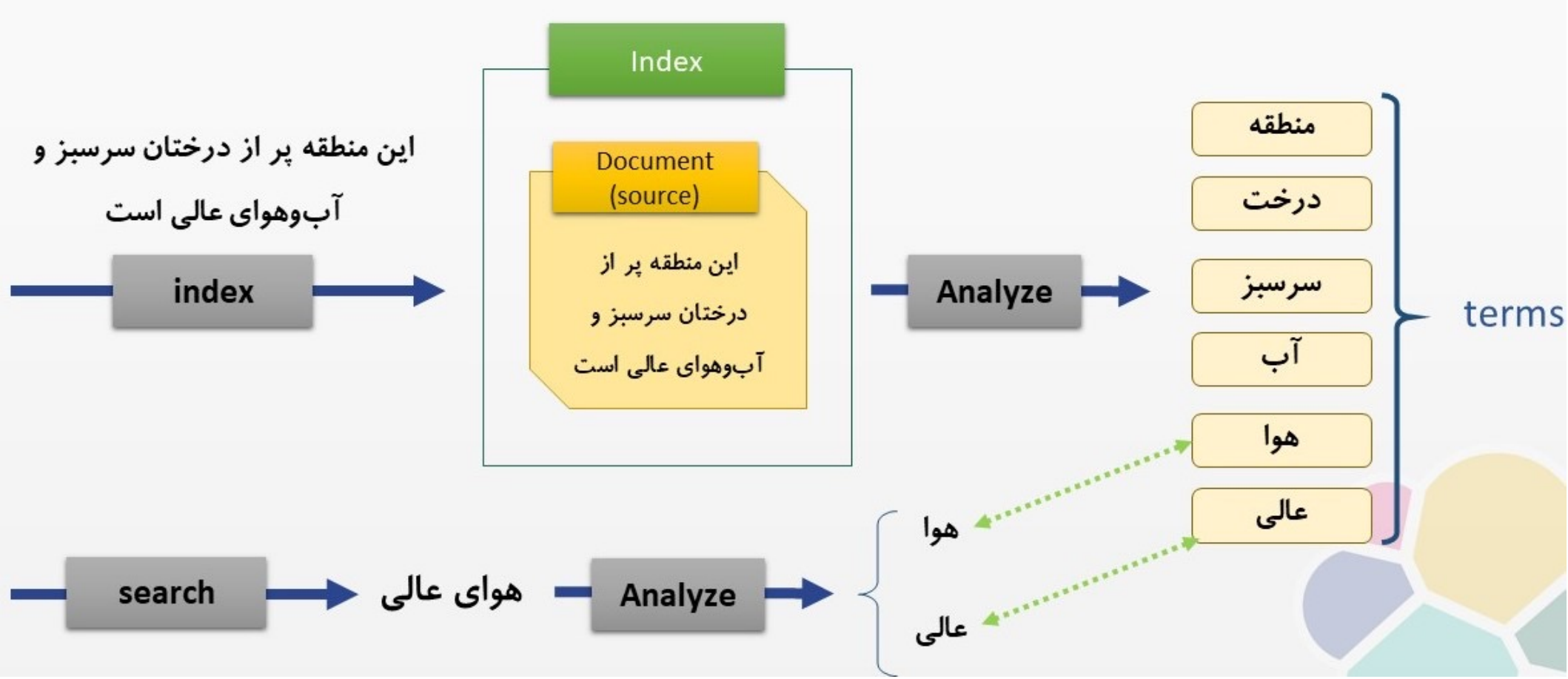
هر Indexمجموعه‌ای از documentهاست و داده هایی که در یک documentذخیره می‌شوند به صورت نیمه ساختاریافته هستند.

زمانیکه یک documentدر Elasticsearchثبت می‌شود، علاوه بر اصل (source)داده های ورودی، بخشی از آن توسط تحلیلگرها (Analyzers)مورد پردازش قرار می گیرد و پس از پردازش به اجزایی کوچک‌تر (terms)تبدیل می‌شود. هدف از پردازش توسط تحلیلگرها، آماده‌سازی داده ها برای شرکت در عملیات جستجو است. شاید بتوان گفت بدون تحلیلگرها، Elasticsearchهیچ حرفی برای گفتن در زمینه‌ی جستجو نداشت!

پس از ثبت داده ها، نوبت به جستجوی آنها می‌رسد. ESقدرت خود در جستجو را، از کتابخانه‌ی Apache Lucene Searchمی‌گیرد. APIهای Elasticsearchبه سادگی به شما اجازه می‌دهند تا Queryهای جستجو را به Elasticsearchارسال کنید.

همچنین تیم Elasticsearchاقدام به انتشار Clientهای مبتنی بر زبان‌های برنامه‌نویسی مختلف کرده‌اند تا به سادگی بتوان در نرم افزارهای گوناگون از APIهای ارائه‌شده‌ی ESاستفاده کرد.

نمای کلی از فرایند تجزیه و تحلیل داده ها



طراحی مبتنی بر اصول "همیشه در دسترس بودن" و "توزیع پذیری"

یکی از نکته های مهم در ، Elasticsearchدرنظر گرفتن طراحی توزیع پذیر (Distributable) آن است به طوری که امکان توسعه ی آن را بسیار ساده و پشتیبانی از حجم بالای داده ها و درخواست ها را امکان پذیر می کند.

افزون بر توزیع پذیری، ساختار Elasticsearchبه گونه ای طراحی شده است که در صورت از دست رفتن قسمتی از منابع مورد استفاده‌ی ES، دسترسی به داده ها تا حد امکان همچنان برقرار باشد. به عبارتی همواره نسخه‌ی کپی از داده ها به صورت توزیع شده در منابع (سرورهای) مختلف وجود داشته باشد تا بتوان از آن‌ها در مواقع نیاز استفاده کرد.

Elasticsearchبر مبنای معماری خوشه بندی (cluster) طراحی شده است. در این معماری مجموعه ای از سرورها که گره نامیده می‌شوند، در یک شبکه ی داخلی به یکدیگر متصل شده و ضمن ارتباط درونی با یکدیگر، درخواست هایی که به سمت آنها ارسال می‌شود را، به گونه ای پاسخ می‌دهند که کاربر تصور می‌کند درخواست ارسال شده با یک سیستم یکپارچه پاسخ داده می شود. به عبارتی در این معماری فرقی نمی‌کند درخواست توسط کدام گره یا سرور دریافت شود و در صورت لزوم گره ها اطلاعاتی را مابین یکدیگر تبادل خواهند کرد و در نهایت پاسخ کاربر داده خواهد شد. بنابراین در Elasticsearchشما به سادگی می‌توانید در صورت نیاز، گرهی جدیدی به خوشه اضافه کنید و Elasticsearchبه طور خودکار بار درخواست‌ها و داده ها را در سطح خوشه تعدیل می‌کند.

برای فهم بهتر مفاهیم فرض کنید لیستی از اطلاعات پروازهای داخلی شرکت هواپیمایی ماهان را به عنوان داده ی نمونه داشته باشیم تا در تعاریف زیر از این داده ی نمونه استفاده کنیم:

Document: یک واحد اطلاعات به فرمت JSON است که در Elasticsearchذخیره می‌شود. در documentها اصل (source) داده ها بدون هیچ تغییری ذخیره می‌شود. در مثال داده ی نمونه، هر documentبرابر اطلاعات یکی از پروازها خواهد بود.

Field :فیلدهای اطلاعاتی یا همان data field ها که به صورت key و value در document ذخیره می‌شوند. برای مثال، ساعت پرواز یا مبدا و مقصد پرواز مثالی هایی از fieldهستند.

Index :به مجموعه‌ای از documentهای مربوط به یک موضوع مشخص، index گفته می‌شود. برای مثال مجموعه ی اطلاعات همه ی پروازها یک index خواهد بود.

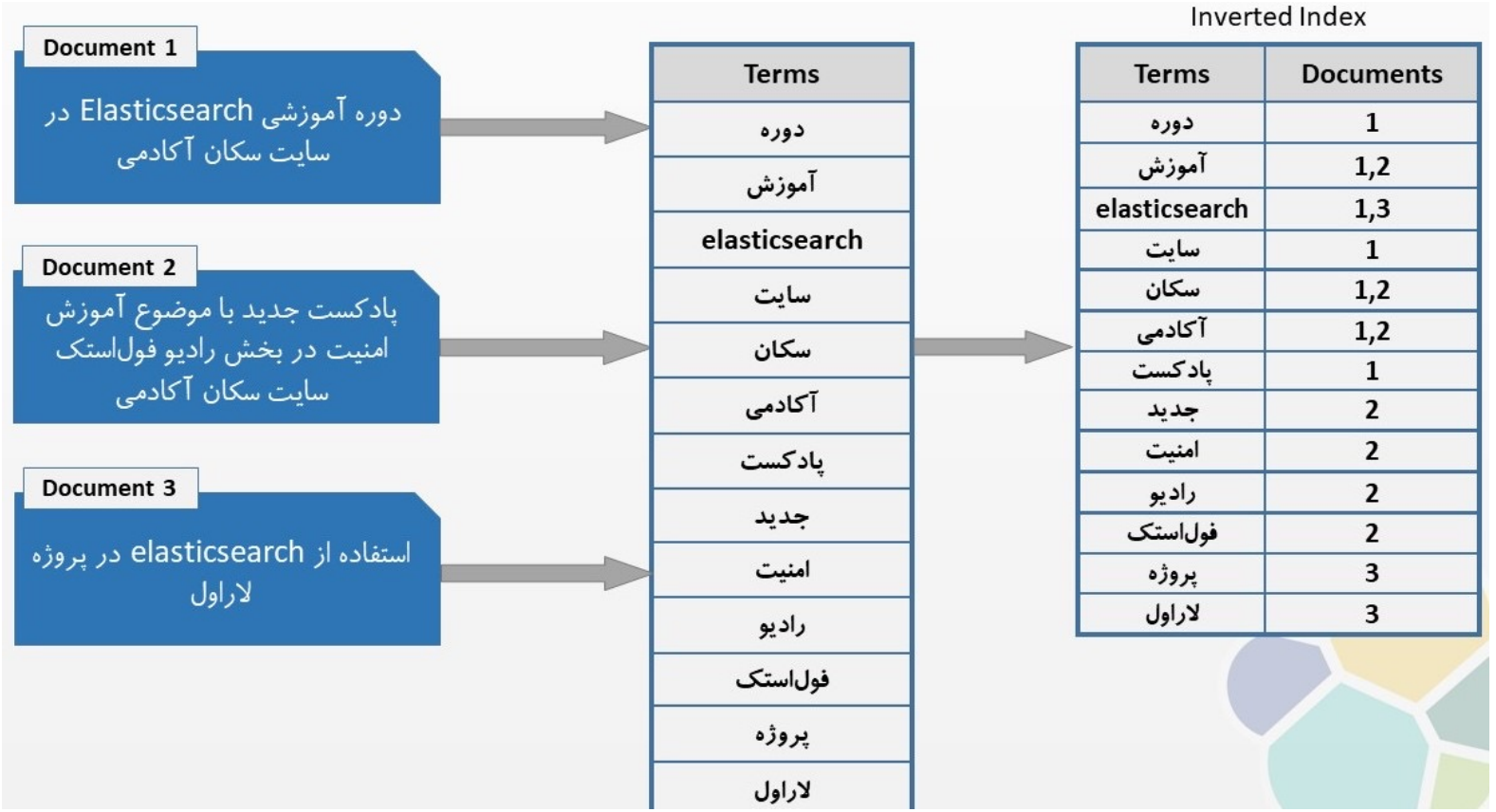
Indexing :عمل ثبت یک documentدر indexرا اصطلاحا indexingیا شاخص‌گذاری می‌گویند. طی این فرایند علاوه بر اصل داده های ،documentکلمات کلیدی متون شناسایی و ذخیره می‌شوند.

Mapping: تعریف ساختار داده ها در فیلدهای اطلاعاتی documentهای یک indexرا mappingمی‌گویند که شامل مواردی همچون نوع داده، نوع تحلیلگر و سایر اطلاعات اضافی در مورد هر یک از fieldها می‌شود. Mappingرا می‌توان مشابه schemaدر پایگاه داده های رابطه ای در نظر گرفت. در نمونه ی داده‌های پرواز، اینکه برای مثال فیلد ساعت پرواز از نوع زمان، شماره پرواز به عنوان کلید یکتا و مبدا و مقصد پرواز از نوع کلمات کلیدی باشند، در واقع mapping ایندکس پروازها را مشخص می‌کند.

Token(Term): هنگامی که یک متن (Text،)توسط تحلیلگر ها در Elasticsearchمورد پردازش قرار گیرد، خروجی آن مجموعه‌ای از termها است که معمولا (نه حتما همیشه!) کلمات مجزا از هم و معنادار بوده و برای شرکت در فرایند جستجو بسیار بهینه هستند. برای مثال عبارت "من یک برنامه نویس حرفه ای هستم" ممکن است تبدیل به termهای { من، برنامه، نویس، حرفه } شود.

Cluster: یک یا مجموعه‌ای از nodeهای Elasticsearchکه با یکدیگر در ارتباط باشند، یک clusterیا خوشه را تشکیل می‌دهد. nodeها در سطح یک cluster با یکدیگر در ارتباط بوده و یکدیگر را می‌شناسند.

Inverted Index : یک ساختارداده‌ی بهینه شده در حافظه رم است که طبق آن به سرعت مشخص می‌شود هر termیا کلمه کلیدی در کدام documentها تکرار شده است. اگر Indexرا همانند فهرست موضوعی ابتدای کتاب‌ها در نظر بگیریم، Inverted Indexمعادل فهرست واژگان انتهای کتاب است که جستجوی سریع یک واژه بر مبنای اینکه در چه صفحه هایی تکرار شده است را ممکن می‌کند!



شمای کلی از ساختار یک خوشه فرضی

- در این خوشه شرایط زیر برقرار است:
- شامل ۲ گره (سرور) به نام‌های node_1 و node_2می‌باشد.
- شامل دو indexبه نام‌های index_1 و index_2 هریک از indexها شامل دو shardمی‌باشد.
- برای index_1هیچگونه replica shard یا نسخه‌ی کپی در نظر گرفته نشده است و برای index_2تعیین شده است تا یک نسخه ی پشتیبان از shardهای اصلی داشته باشد.