Elasticsearch you know for (AI) Search









```
$ curl -XPOST https://localhost:9200/speaker/ doc -d '{
  "name" : "David Pilato",
  "jobs" : [
    { "name" : "SRA Europe (SSII)", "date" : "1995" },
    { "name" : "SFR", "date" : "1997" },
    { "name" : "e-Brands / Vivendi", "date": "2000" },
   { "name" : "DGDDI (douane)", "date" : "2005" },
   { "name" : "elastic", "date" : "2013" }
 ],
  "motivations" : [ "family", "job", "deejay" ],
 "blog" : "https://david.pilato.fr/",
  "x" : [ "@dadoonet", "@elasticfr" ],
  "bluesky" : [ "@pilato.fr" ],
 "email" : "david@elastic.co"
}' -H 'Content-Type: application/json'
```



So Many New Features One Search AI Platform





Elasticsearch — the most widely deployed vector database

Start. Jocal





Elastic Cloud Hosted pricing

Elastic Cloud Hosted is a public cloud managed service available on AWS, Azure, and Google Cloud — delivering search, Al-driven security, and cloud monitoring capabilities.

Want Elastic to manage nodes and shards for you? Explore additional Elastic Cloud offerings.







Gold

As low as **\$114 per month**¹



Everything in Standard plus:

- 🥝 Reporting
- Third-party alerting actions
- 🤣 Watcher
- Multi-stack monitoring

Platinum

As low as **\$131 per month¹**



Everything in Gold plus:

- Advanced Elastic Stack security features
- Machine learning (ML) anomaly detection, supervised learning, third-party model management

Enterprise

As low as **\$184 per month**¹

Try free

Everything in Platinum plus:

- Support for searchable snapshots in cold and frozen tiers
- Elastic Maps Server
- Synthetic _source for storage reduction

Elasticsearch Serverless

Pay only for what you use, with no infrastructure hassle. Discover the art of the possible with AI search, RAG-ready tools, and data analytics capabilities.





A typical search implementation...

```
CREATE TABLE user
(
    name VARCHAR(100),
    comments VARCHAR(1000)
);
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

David



Search on term

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT * FROM user WHERE name="David"; Empty set (0,00 sec)



Ο



Search like

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT * FROM user WHERE name LIKE "%David%";

name	comments
David Pilato	Developer at elastic
David Gageot	Engineer at Doctolib
David David	Who is that guy?

David



 \mathbf{O}

Search for terms

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT * FROM user WHERE name LIKE "%David Pilato%";

++	+
name	comments
David Pilato	Developer at elastic







Search with inverted terms

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT * FROM user WHERE name LIKE "%Pilato David%"; Empty set (0,00 sec) SELECT * FROM user WHERE name LIKE "%Pilato%David%"; Empty set (0,00 sec)







Search for terms

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT * FROM user WHERE name LIKE "%David%" AND
name LIKE "%Pilato%";

name +	comments
David Pilato	Developer at elastic







Search within two fields

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT * FROM user WHERE name LIKE "%David%" OR comments LIKE "%David%";

+	+
name +	comments
David Pilato Malloum Laya David Gageot David David	Developer at elastic Worked with David at french customs service Engineer at Doctolib Who is that guy?



Ο

David



Search with typos

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT * FROM user WHERE name LIKE "%Dadid%"; Empty set (0,00 sec)





Search with typos

```
INSERT INTO user VALUES ('David Pilato', 'Developer at elastic');
INSERT INTO user VALUES ('Malloum Laya', 'Worked with David at french
customs service');
INSERT INTO user VALUES ('David Gageot', 'Engineer at Doctolib');
INSERT INTO user VALUES ('David David', 'Who is that guy?');
```

SELECT	*	FROM	user	WHERE	name	LIKE	"%_adid%"	OR
					name	LIKE	"%D_did%"	OR
					name	LIKE	"%Da_id%"	OR
					name	LIKE	"%Dad_d%"	OR
					name	LIKE	"%Dadi_%"	;

name +	comments
David Pilato	Developer at elastic
David Gageot	Engineer at Doctolib
David David	Who is that guy?

Dadid



Ο



User Interface

Power Search:					
ID Number					
Web Title					
uel					
Category	Select				
Web Description					
Keywords					
Contact Name					
Contact Email					
Featured Links 🐲	Select		۷		
Cool Links 🏎	Select	~			
Bold Links	Select	*			
Icon		0	2		
Rating Average *****	Select	~			
Number of Votes	between			and	
Total Hits	between			and	
Hits Today	between			and	
IP Address					
Submission Software Name					



0

What is a search engine?

Index engine (indexing documents)

faible émissivité, verre 170, 362 faïence, foret 97 faîtage 219 fenêtre 170, 172–177 à vantaux 125, 126 abri de jardin 251 aérateur dynamique 334 barre 132–133 toutes eaux 307 foyer 134, 136 frapper 351 frise carrelage 41 papier peint 87 froid, isolation 330, 363 FSC (Forest Stewardship Council) 20, 69, 202, 361 fuite 299, 309

Search engine (within the created indices)



LANCEZ-VOUS ! TOUT LE SAVOIR-FAIRE DE CASTORAMA







Elasticsearch You know, for search



GET / analyze ł "char filter": ["html strip"], "tokenizer": "standard", "filter": ["lowercase", "stop", "snowball"], "text": "These are not the droids you are looking for."



"char_filter": "html_strip"

These are not the droids you are looking for.



These are not the droids you are looking for.



"tokenizer": "standard"

These are not the droids you are looking for.

These are not the droids are looking for



"filter": "lowercase"

These	these
are	are
not	not
the	the
droids	droids
you	you
are	are
looking	looking
for	for



"filter": "stop"

These	t hese	
are	are	
not	not	
the	the	
droids 🔶	droids	droids
you	you	you
are	are	
looking	looking	looking
for	for	



"filter": "snowball"





```
These are <em>not</em> the droids you are looking for.
{ "tokens": [{
      "token": "droid",
      "start offset": 27, "end offset": 33,
      "type": "<ALPHANUM>", "position": 4
    _} , {
      "token": "you",
      "start offset": 34, "end offset": 37,
      "type": "<ALPHANUM>", "position": 5
    }, {
      "token": "look",
      "start offset": 42, "end offset": 49,
      "type": "<ALPHANUM>", "position": 7
    }]}
```



Elasticsearch You know, for vector search



Embeddings represent your data Example: 1-dimensional vector





Multiple dimensions represent different data aspects





Similar data is grouped together





Vector search ranks objects by similarity (~relevance) to the query



Data Ingestion and Embedding Generation









Similarity





Similarity: cosine (cosine)





https://djdadoo.pilato.fr/







https://github.com/dadoonet/music-search/



So Many New Features One Search AI Platform







www.meetup.com/ElasticFR





discuss.elastic.co





Thank You

