**Python Elasticsearch Solution**

# Install elasicsearch and validate the elasticsearch service

helm repo add elastic <https://helm.elastic.co>

helm install elasticsearch elastic/elasticsearch -f ./values.yaml

kubectl get pods -l app=elasticsearch-master -w

helm test elasticsearch

kubectl port-forward svc/elasticsearch-master 9200

curl -XGET <http://localhost:9200/>

**values.yml**

---

# Permit co-located instances for solitary minikube virtual machines.

antiAffinity: "soft"

# Shrink default JVM heap.

esJavaOpts: "-Xmx128m -Xms128m"

# Allocate smaller chunks of memory per pod.

resources:

requests:

cpu: "100m"

memory: "512M"

limits:

cpu: "1000m"

memory: "512M"

# Request smaller persistent volumes.

volumeClaimTemplate:

accessModes: [ "ReadWriteOnce" ]

storageClassName: "standard"

resources:

requests:

storage: 100M

# Load data

curl -H 'Content-Type: application/json' -XPOST "http://localhost:9200/cities/\_bulk" --data-binary @worldcities\_100records.json

# Copy Python Programs

**es\_health\_local.py**

from elasticsearch import Elasticsearch

es = Elasticsearch([{'host': 'localhost', 'port': 9200}])

def connect\_elasticsearch(\_es):

if \_es.ping():

print('Elasticsearch is UP!')

else:

print('Elasticsearch is DOWN!')

return \_es

if \_\_name\_\_ == '\_\_main\_\_':

status=connect\_elasticsearch(es)

**es\_insert\_local.py**

from operator import index

from random import random

from elasticsearch import Elasticsearch

es = Elasticsearch([{'host': 'localhost', 'port': 9200}])

index="cities"

city = input("Enter the city: ")

country = input("Enter the country: ")

population = input("Enter the population: ")

id=city+country

doc = {

'city': city,

'country': country,

'population': population,

}

def insert\_elasticsearch(\_es,\_index,\_id,\_doc):

ret = \_es.index(index=index, id=\_id, body=\_doc)

return ret

if \_\_name\_\_ == '\_\_main\_\_':

response=insert\_elasticsearch(es,index,id,doc)

print(response)

**es\_update\_local.py**

from elasticsearch import Elasticsearch

es = Elasticsearch([{'host': 'localhost', 'port': 9200}])

index\_name='cities'

def print\_total\_cities(\_es,\_index\_name):

print("Total Cities:")

print(\_es.count(index=\_index\_name))

def get\_input\_city\_population():

\_city = input("Enter the city to get the population: ")

\_new\_population = input("Enter the updated population: ")

return \_city, \_new\_population

def get\_elasticsearch\_id(\_es,\_index\_name,\_matchdoc):

citysearch = \_es.search(index=\_index\_name, body=\_matchdoc)

current\_population\_id = citysearch["hits"]["hits"][0]["\_id"]

print(current\_population\_id)

return current\_population\_id

def update\_elasticsearch(\_es,\_index\_name,\_current\_population\_id,\_updatedoc):

doctype = "\_doc"

ret = \_es.update(index=\_index\_name, doc\_type=doctype, id=\_current\_population\_id, body=\_updatedoc)

print ('Updated Record:', ret)

return ret

if \_\_name\_\_ == '\_\_main\_\_':

print\_total\_cities(es,index\_name)

city, new\_population = get\_input\_city\_population()

matchdoc = {'query': {'bool': {'must': [{'match': {'city': city}}]}}}

body = {'query': {'bool': {'must': [{'match': {'city': city}}]}}}

updatedoc = {

'doc' : {

'population': new\_population

}

}

current\_population\_id = get\_elasticsearch\_id(es,index\_name,matchdoc)

response = update\_elasticsearch(es,index\_name,current\_population\_id,updatedoc)

print(response)

**es\_search\_local.py**

from elasticsearch import Elasticsearch

es = Elasticsearch([{'host': 'localhost', 'port': 9200}])

index\_name='cities'

print("Total Cities:")

print(es.count(index=index\_name))

city = input("Enter the city to get the population: ")

doc = {'query': {'bool': {'must': [{'match': {'city': city}}]}}}

def search\_elasticsearch(\_es,\_index\_name,\_doc):

citysearch = \_es.search(index=\_index\_name, body=\_doc)

population = citysearch["hits"]["hits"][0]["\_source"]["population"]

return population

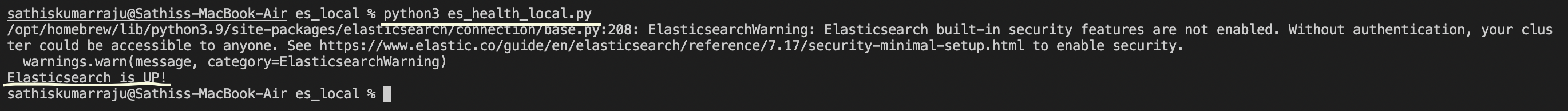
if \_\_name\_\_ == '\_\_main\_\_':

response=search\_elasticsearch(es,index\_name,doc)

print("Population is:", response)

# Run Python programs to interact with elasticsearch

## Health Check

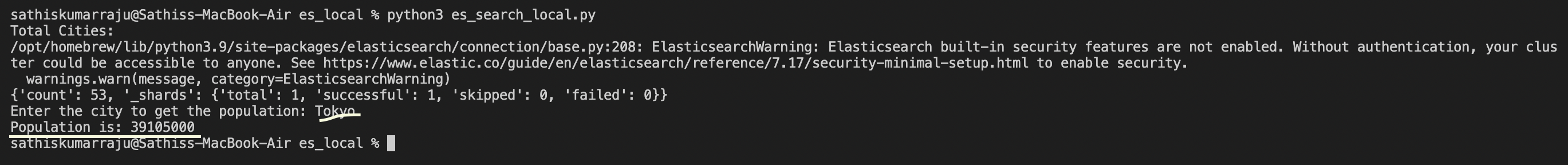


**sathiskumarraju@Sathiss-MacBook-Air es\_local %** python3 es\_health\_local.py

/opt/homebrew/lib/python3.9/site-packages/elasticsearch/connection/base.py:208: ElasticsearchWarning: Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable security.

warnings.warn(message, category=ElasticsearchWarning)

Elasticsearch is UP!



## 2. Search city

**sathiskumarraju@Sathiss-MacBook-Air es\_local %** python3 es\_search\_local.py

Total Cities:

/opt/homebrew/lib/python3.9/site-packages/elasticsearch/connection/base.py:208: ElasticsearchWarning: Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable security.

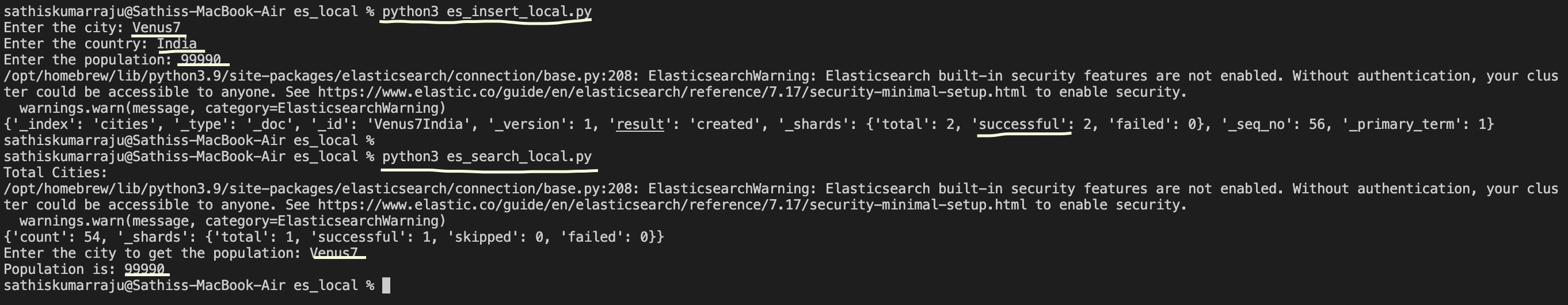
warnings.warn(message, category=ElasticsearchWarning)

{'count': 53, '\_shards': {'total': 1, 'successful': 1, 'skipped': 0, 'failed': 0}}

Enter the city to get the population: Tokyo

Population is: 39105000

## 3. Insert new City, Country and its population



**sathiskumarraju@Sathiss-MacBook-Air es\_local %** python3 es\_insert\_local.py

Enter the city: Venus7

Enter the country: India

Enter the population: 99990

/opt/homebrew/lib/python3.9/site-packages/elasticsearch/connection/base.py:208: ElasticsearchWarning: Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable security.

warnings.warn(message, category=ElasticsearchWarning)

{'\_index': 'cities', '\_type': '\_doc', '\_id': 'Venus7India', '\_version': 1, 'result': 'created', '\_shards': {'total': 2, 'successful': 2, 'failed': 0}, '\_seq\_no': 56, '\_primary\_term': 1}

**sathiskumarraju@Sathiss-MacBook-Air es\_local %** python3 es\_search\_local.py

Total Cities:

/opt/homebrew/lib/python3.9/site-packages/elasticsearch/connection/base.py:208: ElasticsearchWarning: Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable security.

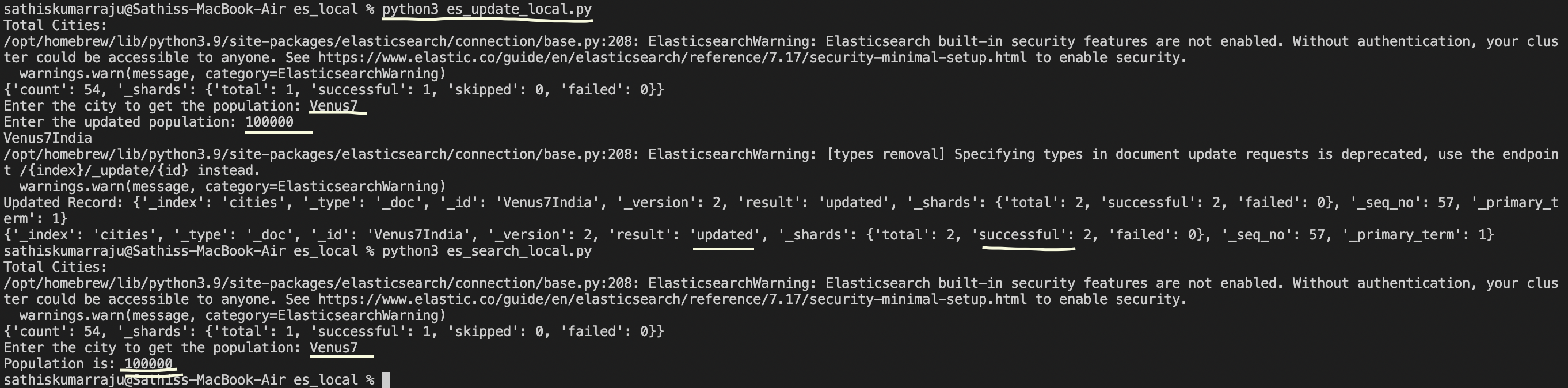
warnings.warn(message, category=ElasticsearchWarning)

{'count': 54, '\_shards': {'total': 1, 'successful': 1, 'skipped': 0, 'failed': 0}}

Enter the city to get the population: Venus7

Population is: 99990

## 4. Update Population count of a City



**sathiskumarraju@Sathiss-MacBook-Air es\_local %** python3 es\_update\_local.py

Total Cities:

/opt/homebrew/lib/python3.9/site-packages/elasticsearch/connection/base.py:208: ElasticsearchWarning: Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable security.

warnings.warn(message, category=ElasticsearchWarning)

{'count': 54, '\_shards': {'total': 1, 'successful': 1, 'skipped': 0, 'failed': 0}}

Enter the city to get the population: Venus7

Enter the updated population: 100000

Venus7India

/opt/homebrew/lib/python3.9/site-packages/elasticsearch/connection/base.py:208: ElasticsearchWarning: [types removal] Specifying types in document update requests is deprecated, use the endpoint /{index}/\_update/{id} instead.

warnings.warn(message, category=ElasticsearchWarning)

Updated Record: {'\_index': 'cities', '\_type': '\_doc', '\_id': 'Venus7India', '\_version': 2, 'result': 'updated', '\_shards': {'total': 2, 'successful': 2, 'failed': 0}, '\_seq\_no': 57, '\_primary\_term': 1}

{'\_index': 'cities', '\_type': '\_doc', '\_id': 'Venus7India', '\_version': 2, 'result': 'updated', '\_shards': {'total': 2, 'successful': 2, 'failed': 0}, '\_seq\_no': 57, '\_primary\_term': 1}

**sathiskumarraju@Sathiss-MacBook-Air es\_local %** python3 es\_search\_local.py

Total Cities:

/opt/homebrew/lib/python3.9/site-packages/elasticsearch/connection/base.py:208: ElasticsearchWarning: Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable security.

warnings.warn(message, category=ElasticsearchWarning)

{'count': 54, '\_shards': {'total': 1, 'successful': 1, 'skipped': 0, 'failed': 0}}

Enter the city to get the population: Venus7

Population is: 100000