CStor Vector python SDK 使用说明文档

下载与安装

资源下载

可在天翼云官网下载，下载地址：[天翼云盘]https://cloud.189.cn/web/share?code=MR3Ub26B3e6f（访问码：yhj7）

环境依赖

python >= 3.9

安装方式

从天翼云官网下载的 SDK 包, 直接解压, 然后进入安装目录:

```
cd vector-python-sdk/
```

安装依赖:

```
pip3 install -r requirements.txt
```

通用：Client 初始化

所有接口都需要初始化 `client`。

```
from boto3.session import Session

url = "your endpoint"
access_key = "your ak"
secret_key = "your sk"
region = "your region"

session = Session(access_key, secret_key)
client = session.client("s3vectors", endpoint_url=url, region_name=region)
```

参数说明

- `access_key` / `secret_key`：认证信息
- `region`：区域
- `endpoint_url`：S3 Vectors 访问地址

---

1. create_vector_bucket

功能

创建向量桶。

输入参数

参数名称	参数含义	参数类型	是否必须
vectorBucketName	需要创建的向量桶名称	string	是

输出参数

- `vectorBucketArn`：向量桶 ARN

代码示例

```
# 第一个示例展示完整的代码，后续示例通用的初始化代码不再展示
from boto3.session import Session

# 通用：Client 初始化
url = "your endpoint"
access_key = "your ak"
secret_key = "your sk"
region = "your region"

session = Session(access_key, secret_key)
client = session.client("s3vectors", endpoint_url=url, region_name=region_name)

# 指定bucket name
vectorBucketName = "your bucket name"

res = client.create_vector_bucket(vectorBucketName=vectorBucketName)
print(res)
```

---

2. delete_vector_bucket

功能

删除向量桶。

输入参数

参数名称	参数含义	参数类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶，以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个

输出参数

- 返回为空对象 `{}`

代码示例

```
resp = client.delete_vector_bucket(
    vectorBucketName="my-vector-bucket"
)
print(resp)
```

---

3. get_vector_bucket

功能

查询向量桶信息。

输入参数

参数名称	参数含义	参数类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个

输出参数

- `vectorBucket.vectorBucketName`：向量桶名称
- `vectorBucket.vectorBucketArn`：向量桶 ARN
- `vectorBucket.creationTime`：创建时间

代码示例

```
resp = client.get_vector_bucket(
    vectorBucketName="my-vector-bucket"
)
print(resp)
```

---

4. list_vector_buckets

功能

列出所有向量桶。

输入参数

参数名称	参数含义	参数类型	是否必须
maxResults	最大返回条数	int	否
prefix	名称前缀过滤	string	否
nextToken	分页token	string	否

输出参数

- `nextToken`：下一页分页 token（如有）
- `vectorBuckets[]`：向量桶列表
- `vectorBuckets[].vectorBucketName`：向量桶名称
- `vectorBuckets[].vectorBucketArn`：向量桶 ARN
- `vectorBuckets[].creationTime`：创建时间

代码示例

```
resp = client.list_vector_buckets(
    maxResults=10,
    prefix="my-",
)
print(resp)
```

---

5. put_vector_bucket_policy

功能

设置向量桶策略。

输入参数

参数名称	参数含义	参数类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
policy	策略 JSON 字符串	string	是

policy参数说明:

字段	描述	类型	是否必须
Version	保持与Amazon S3一致，当前支持"2012-10-17"	string	否
Id	桶策略ID，桶策略的唯一标识	string	否
Statement	桶策略描述，定义完整的权限控制。每条桶策略的Statement可由多条描述组成，每条描述是一个dict，每条描述可包含以下字段： Sid Effect Principal Action ResSource Condition	list	是
Sid	本条桶策略描述的ID	string	否
Effect	桶策略的效果，即指定本条桶策略描述的权限是接受请求还是拒绝请求。 接受请求：配置为“Allow”， 拒绝请求：配置为“Deny”	string	是
Principal	被授权人，即指定本条桶策略描述所作用的用户，支持通配符“*”，表示所有用户。当对某个user进行授权时，Principal格式为: "AWS":"arn:cstor:iam:::user/userId"	dict	否
Action	操作，即指定本条桶策略描述所作用的ZOS操作。以列表形式表示，可配置多条操作，以逗号间隔。支持通配符 “*“，表示该资源能进行的所有操作。Action格式示例为: "zosvectors:GetVectorBucket"，"zosvectors:ListVectorBuckets"	list	否
Resource	指定策略作用的一组资源, 支持用通配 符"*"表示所有资源, 如需要指定具体资源, 则需要在Resource中指定资源的ARN, "arn:cstor:vectors:{region-id}:{acount-id}:bucket/{bucket-name}"	string	否

输出参数

- 返回为空对象 `{}`

代码示例

```
policy_content = '{"Version": "2012-10-17",'\
    '"Id": "id1",' \
        '"Statement": [{' \
        '"Sid": "sid1",' \
        '"Effect": "Allow",' \
        '"Action": "zosvectors:*",' \
        '"Resource": "arn:cstor:vectors:region:myak:bucket/my-vector-bucket",' \
        '"Principal": "*"' \
    '}]}'

resp = client.put_vector_bucket_policy(
    vectorBucketName="my-vector-bucket",
    policy=policy_content
)
print(resp)
```

6. get_vector_bucket_policy

功能

获取向量桶策略。

输入参数

参数名称	参数含义	参数类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个

输出参数

- `policy`：策略 JSON 字符串

代码示例

```
resp = client.get_vector_bucket_policy(
    vectorBucketName="my-vector-bucket",
)
print(resp)
```

---

7. delete_vector_bucket_policy

功能

删除向量桶策略。
-

输入参数

参数名称	参数含义	参数类型	是否必须
ectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个

输出参数

- 返回为空对象 `{}`

代码示例

```
resp = client.delete_vector_bucket_policy(
    vectorBucketName="my-vector-bucket",
)
print(resp)
```

---

8. create_index

功能

在向量桶中创建索引。

输入参数

参数名称	参数含义	参数类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
indexName	索引名称	string	是
dimension	向量维度	int	是
dataType	向量数据类型, 默认float32	string	是
distanceMetric	距离度量, 支持cosine和euclidean	string	是
metadataConfiguration	非过滤元数据键列表	dict	否

输出参数

- `indexArn`：索引 ARN

代码示例

```python
# 创建 index
resp = client.create_index(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    dimension=4,
    dataType="float32",
    distanceMetric="cosine",
    metadataConfiguration={
        "nonFilterableMetadataKeys": ["raw_text"]
    },
)
print(resp)
```

---

9. get_index

功能

查询索引信息。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个

输出参数

- `index.vectorBucketName`：向量桶名称
- `index.indexName`：索引名称
- `index.indexArn`：索引 ARN
- `index.creationTime`：创建时间
- `index.dataType`：向量数据类型（如 `float32`）
- `index.dimension`：向量维度
- `index.distanceMetric`：距离度量（`cosine` 或 `euclidean`）
- `index.metadataConfiguration.nonFilterableMetadataKeys[]`：不可过滤元数据键列表

代码示例

```python
resp = client.get_index(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
)
print(resp)
```

---

10. list_indexes

功能

列出向量桶内的索引。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
maxResults	最大返回索引数量	int	是
prefix	索引名称前缀	string	否
nextToken	分页 token	string	否

输出参数

- `nextToken`：下一页分页 token（如有）
- `indexes[]`：索引列表
- `indexes[].vectorBucketName`：向量桶名称
- `indexes[].indexName`：索引名称
- `indexes[].indexArn`：索引 ARN
- `indexes[].creationTime`：创建时间

代码示例

```
resp = client.list_indexes(
    vectorBucketName="my-vector-bucket",
    maxResults=10,
    prefix="demo-",
)
print(resp)
```

---

11. delete_index

功能

删除索引。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
indexName	索引名称	string	indexName或indexArn必须指定一个
indexArn	索引ARN	string	indexName或indexArn必须指定一个

输出参数

- 返回为空对象 `{}`

代码示例

```
resp = client.delete_index(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
)
print(resp)
```

---

12. put_vectors

功能

插入或更新向量。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
indexName	索引名称	string	indexName或indexArn必须指定一个
indexArn	索引ARN	string	indexName或indexArn必须指定一个
vectors	向量数组	list	是

输出参数

- 返回为空对象 `{}`

代码示例 （含 bucket/index 初始化）

```
client.create_vector_bucket(vectorBucketName="my-vector-bucket")
client.create_index(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    dimension=4,
    dataType="float32",
    distanceMetric="cosine",
)

vectors = [
    {
        "key": "doc-001",
        "data": {"float32": [0.1, 0.2, 0.3, 0.4]},
        "metadata": {"title": "Doc 1", "category": "A"},
    }
]

resp = client.put_vectors(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    vectors=vectors,
)
print(resp)
```

---

13. get_vectors

功能

获取指定向量。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
indexName	索引名称	string	indexName或indexArn必须指定一个
indexArn	索引ARN	string	indexName或indexArn必须指定一个
keys	向量 key 列表	list	是
returnData	是否返回向量数据	bool	否
returnMetadata	是否返回元数据	bool	否

输出参数

- `vectors[]`：向量列表
- `vectors[].key`：向量 key
- `vectors[].data.float32[]`：向量数据（当 `returnData=True`）
- `vectors[].metadata`：元数据（当 `returnMetadata=True`）

代码示例（含 bucket/index 初始化）

```
client.create_vector_bucket(vectorBucketName="my-vector-bucket")
client.create_index(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    dimension=4,
    dataType="float32",
    distanceMetric="cosine",
)

client.put_vectors(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    vectors=[
        {"key": "doc-001", "data": {"float32": [0.1, 0.2, 0.3, 0.4]}}
    ],
)

resp = client.get_vectors(
```

```
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    keys=["doc-001"],
    returnData=True,
    returnMetadata=True,
)
print(resp)
```

---

14. list_vectors

功能

列出索引中的向量列表。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
indexName	索引名称	string	indexName或indexArn必须指定一个
indexArn	索引ARN	string	indexName或indexArn必须指定一个
maxResults	最大返回结果数	int	否
nextToken	分页 token	string	否
returnData	是否返回向量数据	bool	否
returnMetadata	是否返回元数据	bool	否
segmentCount	分片总数	int	否
segmentIndex	分片索引	int	否

输出参数

- `nextToken`：下一页分页 token（如有）
- `vectors[]`：向量列表
- `vectors[].key`：向量 key
- `vectors[].data.float32[]`：向量数据（当 `returnData=True`）
- `vectors[].metadata`：元数据（当 `returnMetadata=True`）

代码示例（含 bucket/index 初始化）

```
client.create_vector_bucket(vectorBucketName="my-vector-bucket")
client.create_index(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    dimension=4,
    dataType="float32",
    distanceMetric="cosine",
)

resp = client.list_vectors(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    maxResults=10,
    returnMetadata=True,
)
print(resp)
```

---

15. delete_vectors

功能

删除指定向量。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
indexName	索引名称	string	indexName或indexArn必须指定一个
indexArn	索引ARN	string	indexName或indexArn必须指定一个
keys	向量 key 列表	list	是

输出参数

- 返回为空对象 `{}`

代码示例（含 bucket/index 初始化）

```
client.create_vector_bucket(vectorBucketName="my-vector-bucket")
client.create_index(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    dimension=4,
    dataType="float32",
    distanceMetric="cosine",
)

client.put_vectors(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    vectors=[
        {"key": "doc-001", "data": {"float32": [0.1, 0.2, 0.3, 0.4]}}
    ],
)

resp = client.delete_vectors(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    keys=["doc-001",],
)
print(resp)
```

---

16. query_vectors

功能

向量相似度搜索。

输入参数

参数名称	参数含义	类型	是否必须
vectorBucketName	需要操作的向量桶名称	string	vectorBucketName和vectorBucketArn只能指定一个
vectorBucketArn	需要操作的向量桶, 以arn格式指定(arn在创建桶时返回)	string	vectorBucketName和vectorBucketArn只能指定一个
indexName	索引名称	string	indexName或indexArn必须指定一个
indexArn	索引ARN	string	indexName或indexArn必须指定一个
queryVector	查询向量	dict	是
topK	返回结果数	int	是
filter	过滤条件 JSON	dict	否
returnDistance	是否返回距离值	bool	否
returnMetadata	是否返回元数据	bool	否

输出参数

- `vectors[]`：相似度搜索结果
- `vectors[].key`：向量 key
- `vectors[].distance`：距离值（当 `returnDistance=True`）
- `vectors[].metadata`：元数据（当 `returnMetadata=True`）
- `distanceMetric`：距离度量（`cosine` 或 `euclidean`）

代码示例

```
query_vector = [0.1, 0.2, 0.3, 0.4]

resp = client.query_vectors(
    vectorBucketName="my-vector-bucket",
    indexName="my-index",
    queryVector={"float32": query_vector},
    topK=5,
    returnDistance=True,
    returnMetadata=True,
)
print(resp)
```