CLOUD.GOV

GSA

AWS Redis & Elasticsearch Broker

Overview

- Redis service changes
- Elasticsearch service changes
- How to get access?

What’s different with the new Redis service?

The current service is a Cloud.gov managed **Redis** in **Kubernetes**

- Available **Redis v3.2**
- **Micro**, **standard**, and **standard-ha** plans
- Encryption **at-rest** and **Redis AUTH**
- **No automated backups** or data retention

The new service will now be **AWS ElastiCache Redis** instances

- Available **Redis v5**
- **1**, **3**, and **5 node cluster** plans
- Encryption **in-transit**, **at-rest**, and **Redis AUTH**
- **Automated daily backups** with a **3 day retention** period

The improved **data retention**

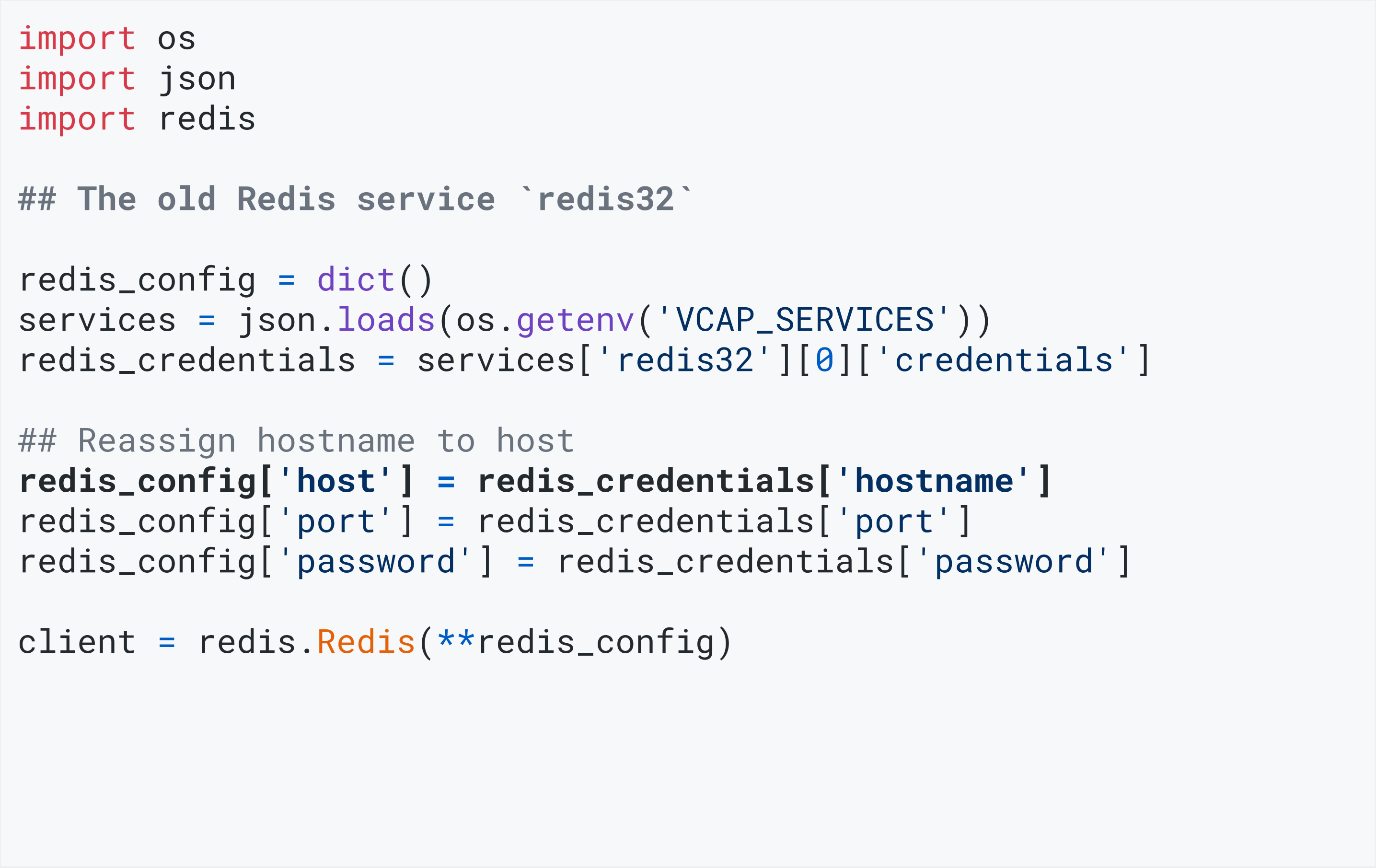
- Before, users had to **roll their own** data restore
- The new **1 node** plan is for development and provides a **daily snapshot**
- The new **3** and **5 node** plans distributes the cluster across **Multi-AZs**, provides **daily snapshots,** and minimizes possible data loss with the **replica nodes**.

How to migrate your service

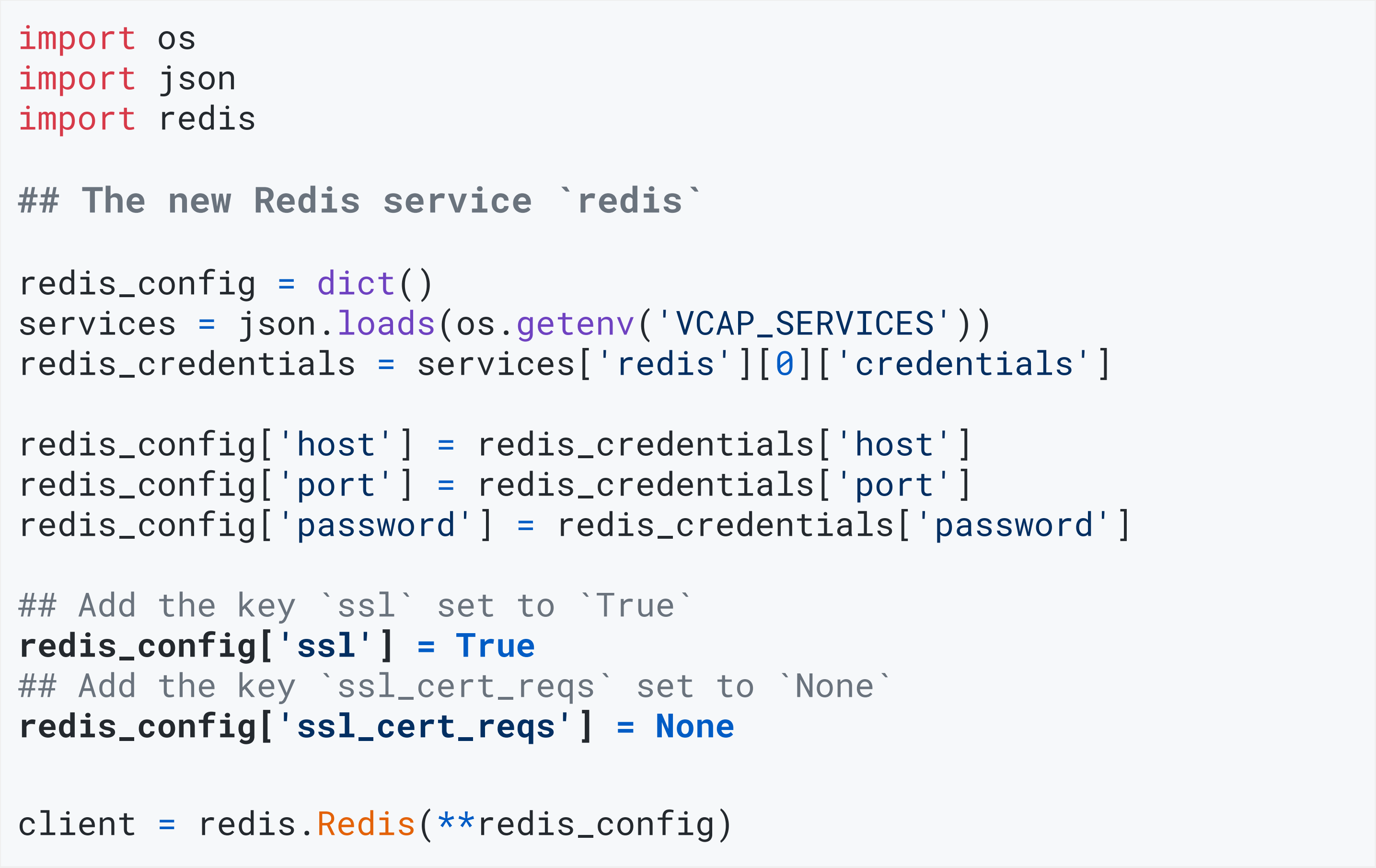
What is the **difference**

- The service name changed from **redis32** to **redis**
- Since the new Redis service uses **in-transit encryption**, you will need to configure your client to support **TLS**
- The service credentials has added **host** and keeps **hostname** for backward compatibility

An example in **Python**



```
import os
import json
import redis

## The old Redis service `redis32`

redis_config = dict()
services = json.loads(os.getenv('VCAP_SERVICES'))
redis_credentials = services['redis32'][0]['credentials']

## Reassign hostname to host
redis_config['host'] = redis_credentials['hostname']
redis_config['port'] = redis_credentials['port']
redis_config['password'] = redis_credentials['password']

client = redis.Redis(**redis_config)
```



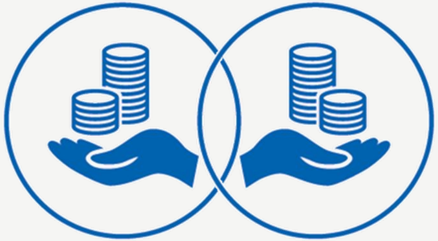
```python
import os
import json
import redis

## The new Redis service `redis`

redis_config = dict()
services = json.loads(os.getenv('VCAP_SERVICES'))
redis_credentials = services['redis'][0]['credentials']

redis_config['host'] = redis_credentials['host']
redis_config['port'] = redis_credentials['port']
redis_config['password'] = redis_credentials['password']

## Add the key `ssl` set to `True`
redis_config['ssl'] = True
## Add the key `ssl_cert_reqs` set to `None`
redis_config['ssl_cert_reqs'] = None


client = redis.Redis(**redis_config)
```

Rule of thumb for handling **SSL/TLS** with your Redis client

- Set the **SSL** (or equivalent) config key to **True**
- Set you **TLS/SSL params** (or equivalent) config to be an **empty** data type **

*** AWS' current implementation of SSL on Redis nodes uses AWS Public CA signed certs that match the background service node names and not the cluster name. This will cause a traditional SSL name mismatch between your cluster URI hostname and the presented certificate. AWS is working to resolve this but no known ETA has been provided by AWS at this time.*

What’s different with the new Elasticsearch service?



The current service is a Cloud.gov managed **Elasticsearch** in **Kubernetes**

- Available **Elasticsearch v5.6**
- **Medium** and **Medium-ha** plans
- Open source S3 snapshot plugin
- **No automated backups** or data retention

The new service will now be **AWS Elasticsearch** instances

- Available **Elasticsearch v7.X**
- **1**, **5**, and **7 node cluster** plans
- Encryption **in-transit**, **between nodes**, **at-rest**, and **AWS IAM based signing requests.**
- **Automated hourly backups** with a **14 day retention** period.

The improved **data retention**

- Before, users had to **roll their own** data restore now it's build in. **Note:** These backups are for cluster restore only and can't be used for data export.
- The new **1 node** plan is for development only.
- The new **5** and **7 node** plans distributes the cluster across **Multi-AZs**, include **3 Master Nodes** and **2 or 4 Data Nodes.**

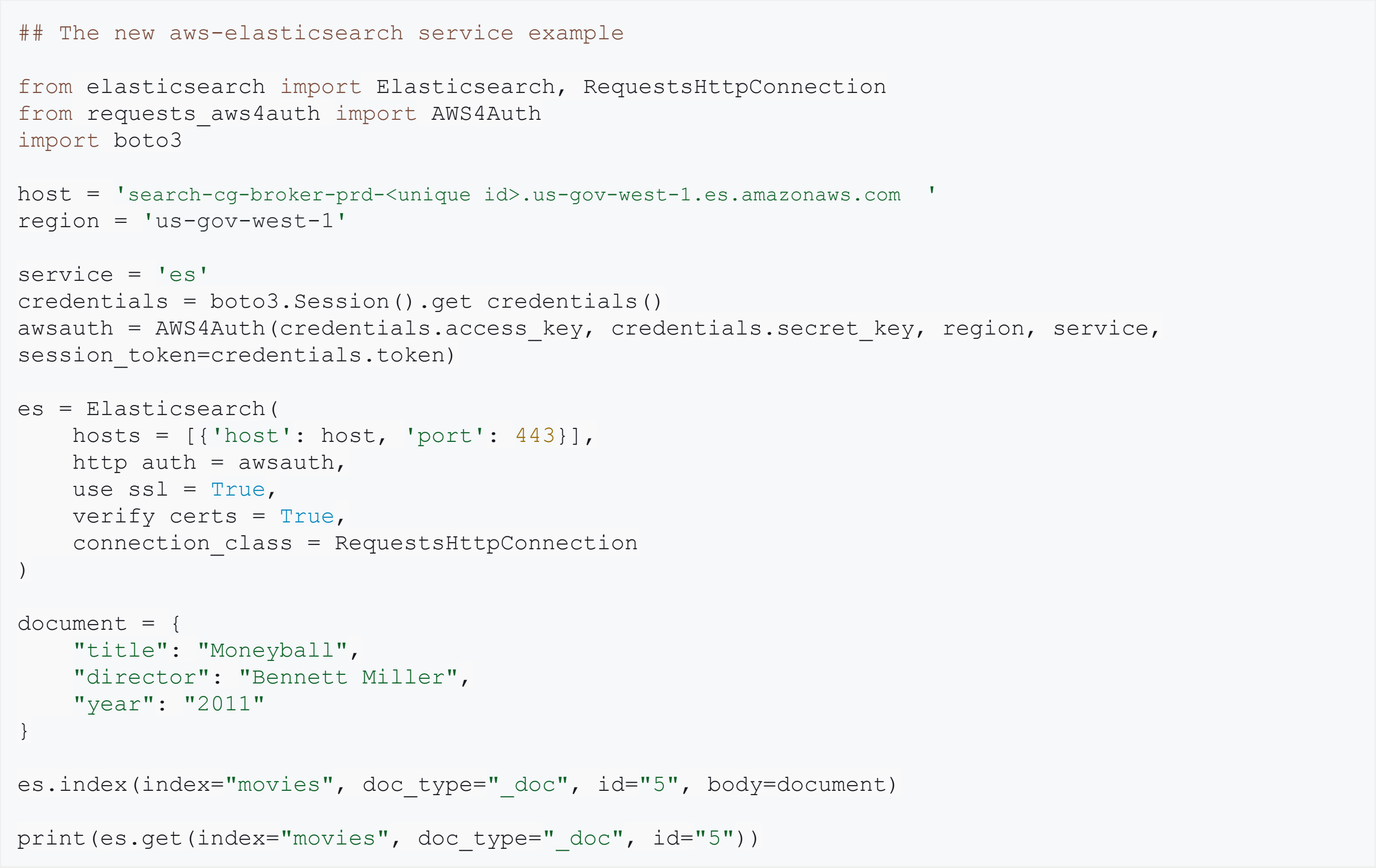
How to migrate your service

What is the **difference**

- The service name changed from **elasticsearch56** to **aws-elasticsearch**
- Since the new ES service uses **AWS IAM Signed requests**, you will need to configure your client to AWS signing.
- https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-request-signing.html

An example in **Python**

```python
## The legacy elasticsearch56 service example

from elasticsearch import Elasticsearch

uri = 'http://username:password.service.kubernetes:9200'

es = Elasticsearch(hosts = [uri])

document = {
    "title": "Moneyball",
    "director": "Bennett Miller",
    "year": "2011"
}

es.index(index="movies", doc_type="_doc", id="5", body=document)

print(es.get(index="movies", doc_type="_doc", id="5"))
```



```
## The new aws-elasticsearch service example

from elasticsearch import Elasticsearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3

host = 'search-cg-broker-prd-<unique id>.us-gov-west-1.es.amazonaws.com  '
region = 'us-gov-west-1'

service = 'es'
credentials = boto3.Session().get credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
session_token=credentials.token)

es = Elasticsearch(
    hosts = [{'host': host, 'port': 443}],
    http auth = awsauth,
    use ssl = True,
    verify certs = True,
    connection_class = RequestsHttpConnection
)

document = {
    "title": "Moneyball",
    "director": "Bennett Miller",
    "year": "2011"
}

es.index(index="movies", doc_type="_doc", id="5", body=document)

print(es.get(index="movies", doc_type="_doc", id="5"))
```

Some **final notes** on Elasticsearch

- cloud.gov is working on manual snapshot implementation for the ES service.
- This will allow customers to export data to S3 as well as import data into the new service
- **Note:** Customers wishing to import their legacy 5.6 data into the new service will need to go through a transitional service offering and then have that instance upgraded since ES does not support snapshot imports greater than a single major version.

How to get access

- Ready to get started? Email support@cloud.gov and we can enable the new Redis and Elasticsearch services for you Cloud.gov organizations
- More documentation and examples are available in our example Github Repositories for each service:
  - Redis: https://github.com/cloud-gov/aws-redis-example
  - Elasticsearch: https://github.com/cloud-gov/aws-elasticsearch-example