# **AWS Academy Cloud Developing [98590]**

<https://awsacademy.instructure.com/courses/98590/assignments/1072469?module_item_id=9121923>

<https://labs.vocareum.com/web/3519877/3539895.0/ASNLIB/public/docs/lang/en-us/README.html>

# Lab 9.1: Caching Application Data with ElastiCache

## **Tổng quan và mục tiêu của bài lab**

Triển khai một cụm Amazon ElastiCache, và kiểm tra việc đồng bộ hóa bộ nhớ đệm với cơ sở dữ liệu Amazon Aurora Serverless bằng Python.

Cập nhật ứng dụng nhà cung cấp cà phê với code Node.js mới, sử dụng bộ nhớ đệm dữ liệu.

## **Sau khi hoàn thành bài lab này, bạn sẽ có thể:**

* Tạo một cụm ElastiCache cho Memcached mới.
* Truy vấn cụm ElastiCache cho Memcached bằng cách sử dụng Python và client pymemcache.

## **Giới thiệu**

### **Amazon ElastiCache**

Amazon ElastiCache là một dịch vụ quản lý bộ nhớ đệm của AWS, giúp triển khai, vận hành và mở rộng các bộ nhớ đệm in-memory trong đám mây. Nó hỗ trợ hai engine phổ biến là Redis và Memcached.

*Những điểm nổi bật của Amazon ElastiCache:*

* **Hiệu năng cao**: Vì là bộ nhớ đệm in-memory, ElastiCache cung cấp tốc độ truy xuất dữ liệu rất nhanh, giúp giảm độ trễ của ứng dụng.
* **Dễ dàng triển khai và quản lý**: ElastiCache tự động xử lý các tác vụ như cấu hình, vá lỗi, sao lưu và khôi phục, giúp bạn tập trung vào phát triển ứng dụng hơn là quản lý hạ tầng.
* **Khả năng mở rộng**: ElastiCache dễ dàng mở rộng và co giãn tài nguyên để đáp ứng nhu cầu lưu trữ và truy xuất dữ liệu tăng cao.
* **Tích hợp chặt chẽ với các dịch vụ AWS khác**: ElastiCache có thể tích hợp tốt với các dịch vụ khác của AWS như Amazon EC2, Amazon RDS, và Amazon DynamoDB, giúp tối ưu hóa hiệu năng tổng thể của hệ thống.

*Sử dụng Amazon ElastiCache trong thực tế:*

ElastiCache thường được sử dụng để cải thiện hiệu năng của các ứng dụng web và di động bằng cách lưu trữ các dữ liệu tạm thời, phiên làm việc (session), và các kết quả truy vấn dữ liệu. Điều này giúp giảm tải trên cơ sở dữ liệu chính và tăng tốc độ phản hồi của ứng dụng.

### **Amazon Aurora Serverless**

Amazon Aurora Serverless là một dịch vụ đám mây từ Amazon Web Services (AWS) cho phép bạn tối ưu hóa chi phí và hiệu suất cho các ứng dụng quản trị cơ sở dữ liệu. Nó tự động điều chỉnh tài nguyên dựa trên nhu cầu thực tế, giúp bạn không phải lo lắng về việc quản lý và mở rộng cơ sở dữ liệu.

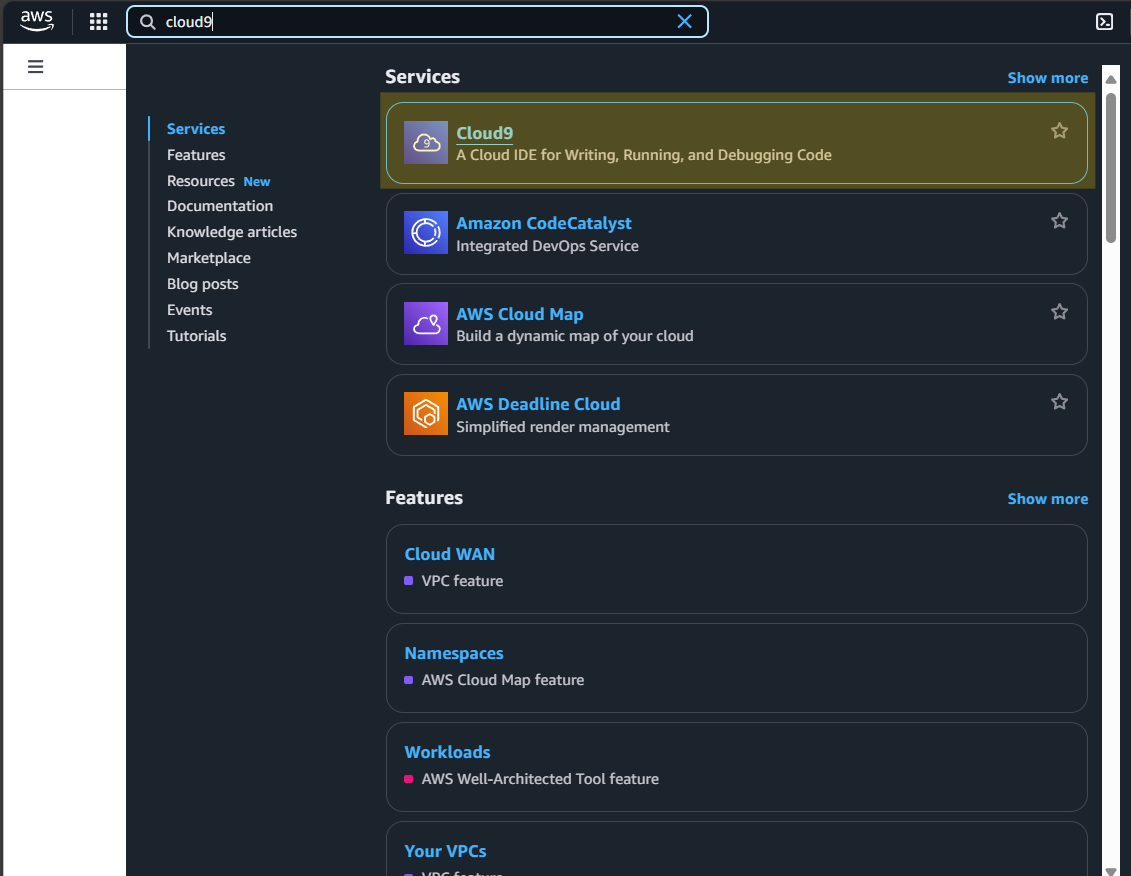
*Một số tính năng nổi bật của Amazon Aurora Serverless bao gồm:*

* **Tính năng tự động**: Tăng cường và giảm thiểu tài nguyên dựa trên nhu cầu thực tế.
* **Chi phí hiệu quả**: Chỉ phải trả tiền cho những tài nguyên bạn thực sự sử dụng.
* **Khả năng mở rộng**: Dễ dàng mở rộng và tối ưu hóa hiệu suất mà không cần điều chỉnh thủ công.

## **Task 1: Preparing the development environment**

Cấu hình môi trường AWS Cloud9 để sử dụng Python và AWS CLI để tương tác với các dịch vụ AWS.

### **5. Mở Service Cloud9**



Bấm Open:

A screenshot of a computer

Description automatically generated

### **6. Tải xuống và giải nén các tập tin bạn cần cho bài thực hành này.**

Chạy lệnh sau trên terminal:

|  |
| --- |
| wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/08-lab-db-caching/code.zip -P /home/ec2-user/environment |

A screenshot of a computer

Description automatically generated

|  |
| --- |
| unzip code.zip |

A screenshot of a computer

Description automatically generated

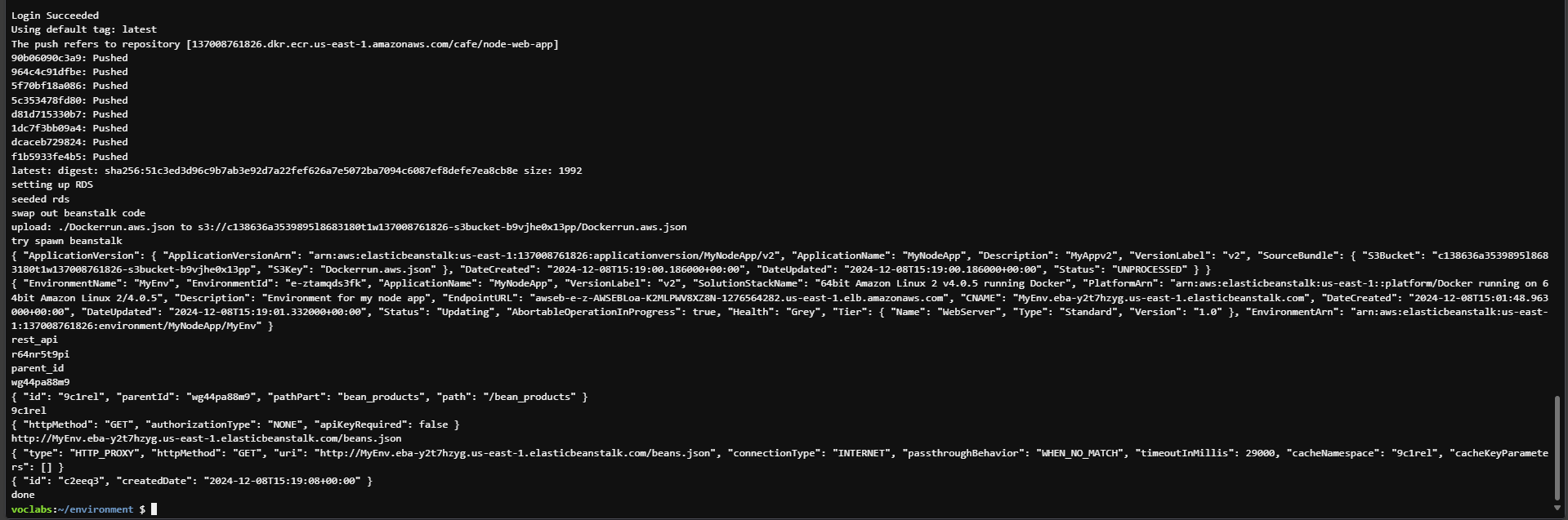
### **7. Chạy tập lệnh để nâng cấp phiên bản Python và AWS CLI được cài đặt trên phiên bản Cloud9.**

Nó cũng sẽ tạo lại công việc mà bạn đã hoàn thành trong các lab trước đó vào tài khoản AWS này.

Chạy lệnh sau trên terminal:

|  |
| --- |
| chmod +x ./resources/setup.sh && ./resources/setup.sh |

Nhập Ipv4 lấy tại <https://whatismyipaddress.com/> khi được yêu cầu.



### **8. Kiểm tra phiên bản AWS CLI**

Chạy lệnh sau trên terminal:

|  |
| --- |
| aws --version |

A screen shot of a computer

Description automatically generated

### **9. Kiểm tra SDK đã cài Python chưa**

|  |
| --- |
| pip show boto3 |

A black background with white text

Description automatically generated

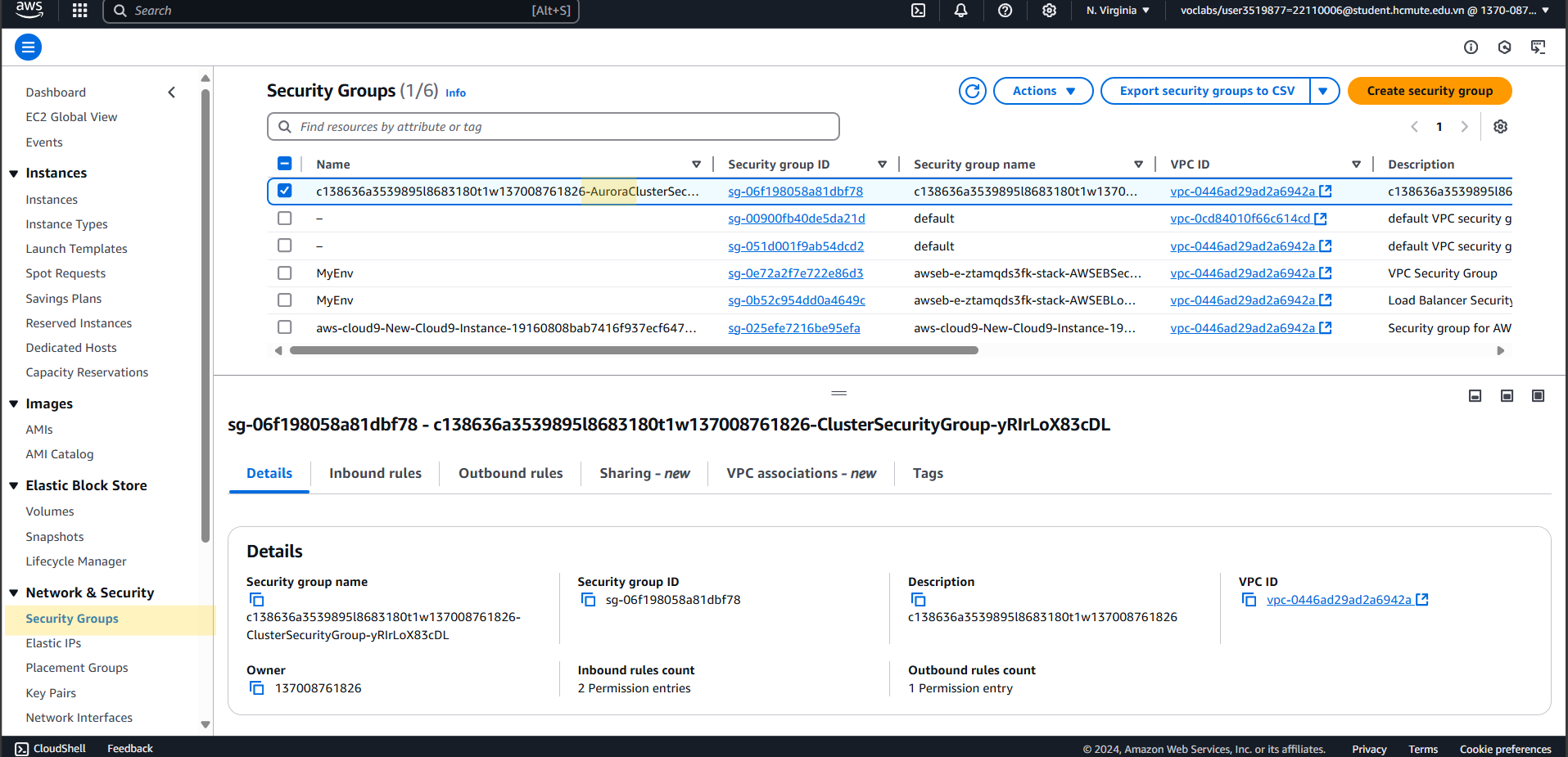
## **Task 2: Configuring the subnets for ElastiCache to use**

CSDL Memcached phải có khả năng giao tiếp với CSDL Aurora Serverless. Ngoài ra, ứng dụng cần có khả năng truy cập cả cơ sở dữ liệu Memcached và CSDL Aurora Serverless.

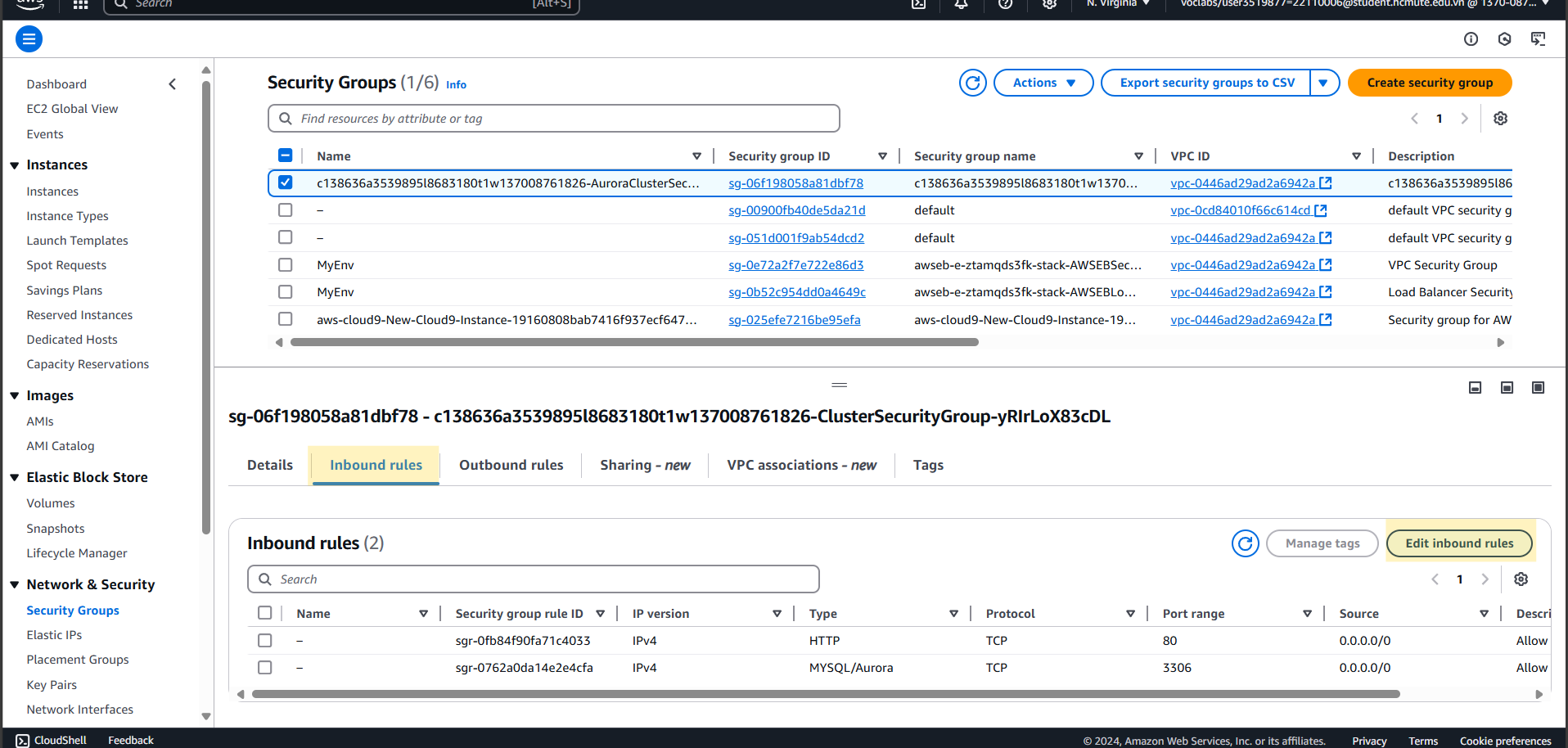
Cần cấu hình một nhóm bảo mật nhằm kích hoạt hoạt động liên lạc cần thiết giữa ứng dụng, cơ sở dữ liệu và bộ nhớ cache. Bạn sẽ sử dụng nhóm bảo mật mà cụm Aurora Serverless đang sử dụng thay vì tạo nhóm bảo mật bổ sung.

### **10. Cấu hình nhóm bảo mật mà Aurora Serverless sẽ sử dụng:**

- Mở service **EC2 🡪 Security Groups 🡪 Chọn nhóm bảo mật có tên Aurora:**



- Chọn **Inbound rules 🡪 Edit Inbound rules:**



- Thêm rule cho port của Memcached:

Bấm Add Rule, điền các thông tin sau:

* **Type:** Custom TCP
* **Port range:** 11211
* **Source**: Chọn nhóm bảo mật có tên chứa "Aurora". Đây là cùng nhóm mà bạn đang chỉnh sửa.

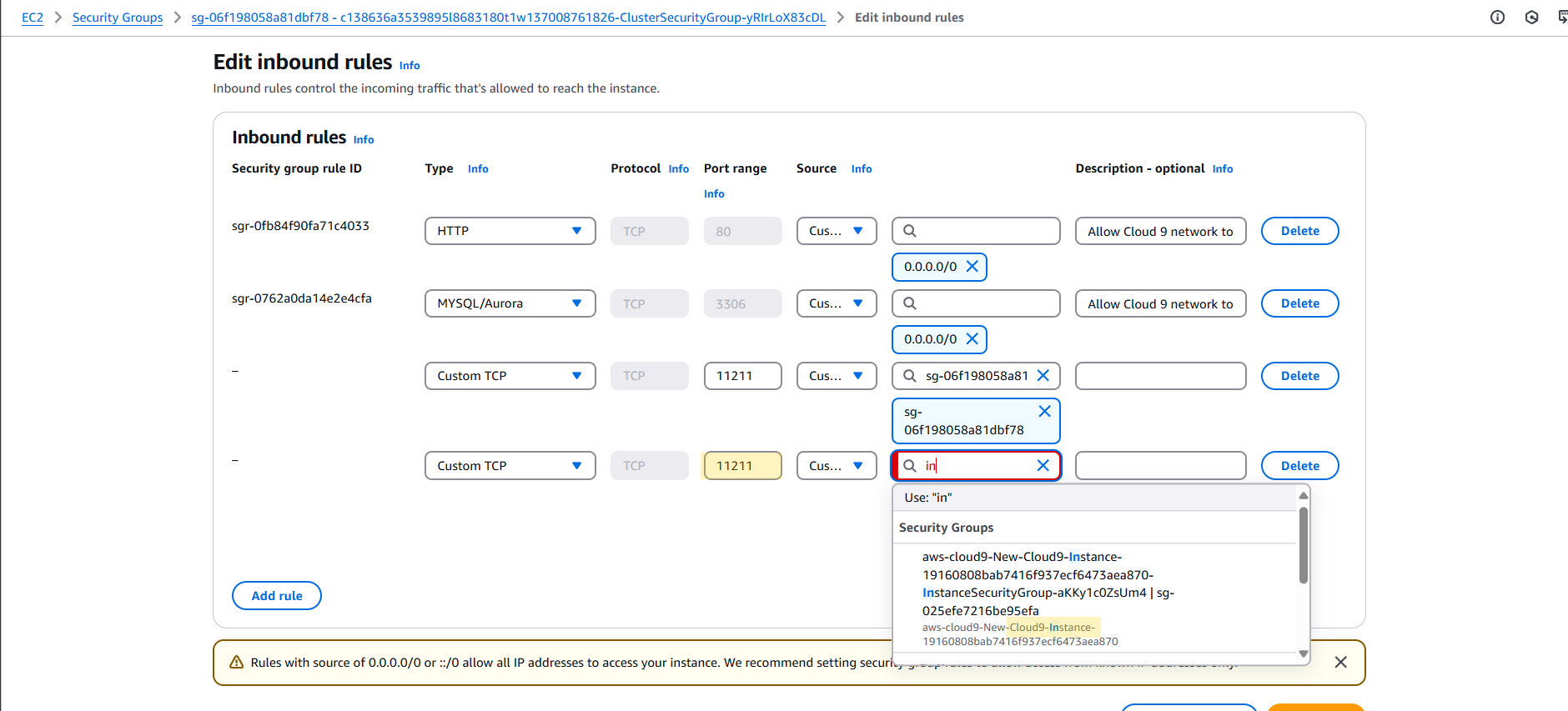
A screenshot of a computer

Description automatically generated

- Thêm quy tắc cho phép truy cập vào cổng Memcached từ tất cả các nguồn được chỉ định cho nhóm bảo mật Cloud9-Instance:

Bấm Add Rule, điền các thông tin sau:

* **Type:** Custom TCP
* **Port range:** 11211
* **Source**: Chọn nhóm bảo mật có tên chứa "Cloud9-Instance".



- Bấm Save Rules.

### **11. Cấu hình nhóm mạng con CSDL mà Aurora Serverless sẽ sử dụng**

- Mở Service **ElastiCache 🡪 Subnet Groups 🡪 Create Subnet Group**



- Điền các thông tin sau:

* **Name:** ElastiCacheSubnetGroup
* **Description:** Subnet Group for ElastiCache
* **VPC ID:** Cloud9 VPC

A screenshot of a computer

Description automatically generated

- Thiết lập Availability Zone:

Mặc định ta có 2 AZ như sau:

A close up of a white background

Description automatically generated

- Bấm Create ở cuối trang để tạo mới subnet group.

Kết quả:

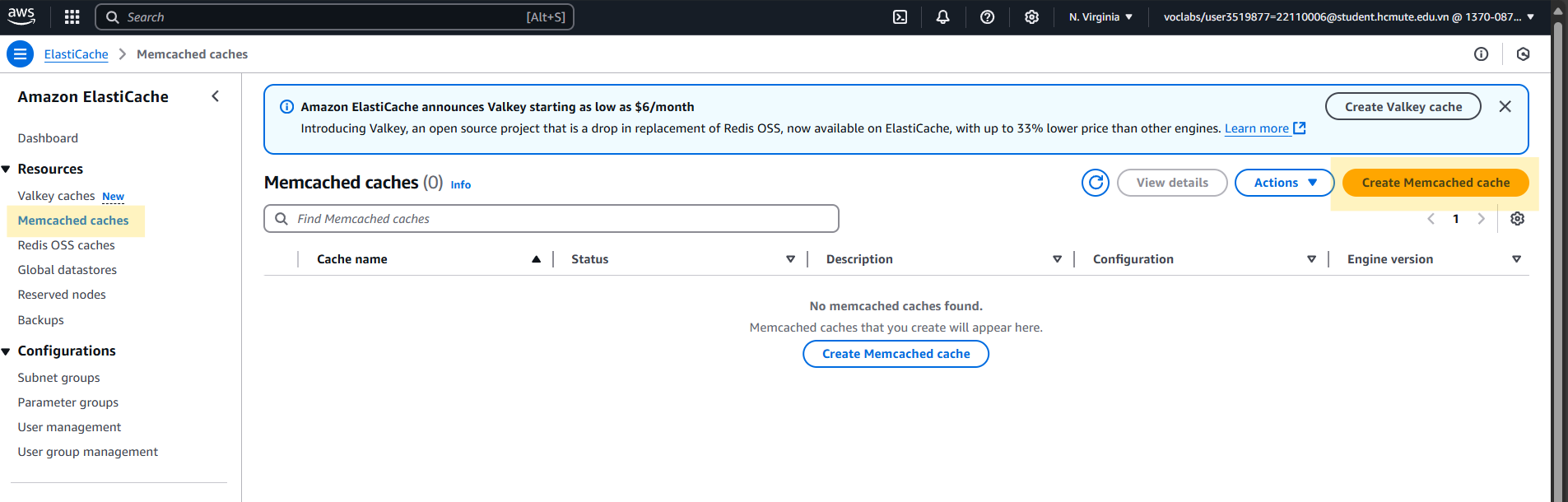
A screenshot of a computer

Description automatically generated

## **Task 3: Creating the ElastiCache cluster**

### **12. Tạo ElastiCache cho phiên bản Memcached.**

- Quay lại service **ElastiCache** 🡪 **Memcached caches 🡪 Create Memcached cache:**



- Chọn các Options như hình:

A screenshot of a computer

Description automatically generated

- Điền các thông tin Cluster như sau:

* **Name:** MemcachedCache
* **Engine version compatibility:** 1.6.6
* **Port:** 11211
* **Parameter group:** default.memcached.1.6
* **Node type:** cache.r6g.large (13.07GiB)
* **Number of nodes:** 3

A screenshot of a computer

Description automatically generated

- Cài đặt **Connectivity:**

A screenshot of a computer

Description automatically generated

- Cài đặt **Availability zones placement:**

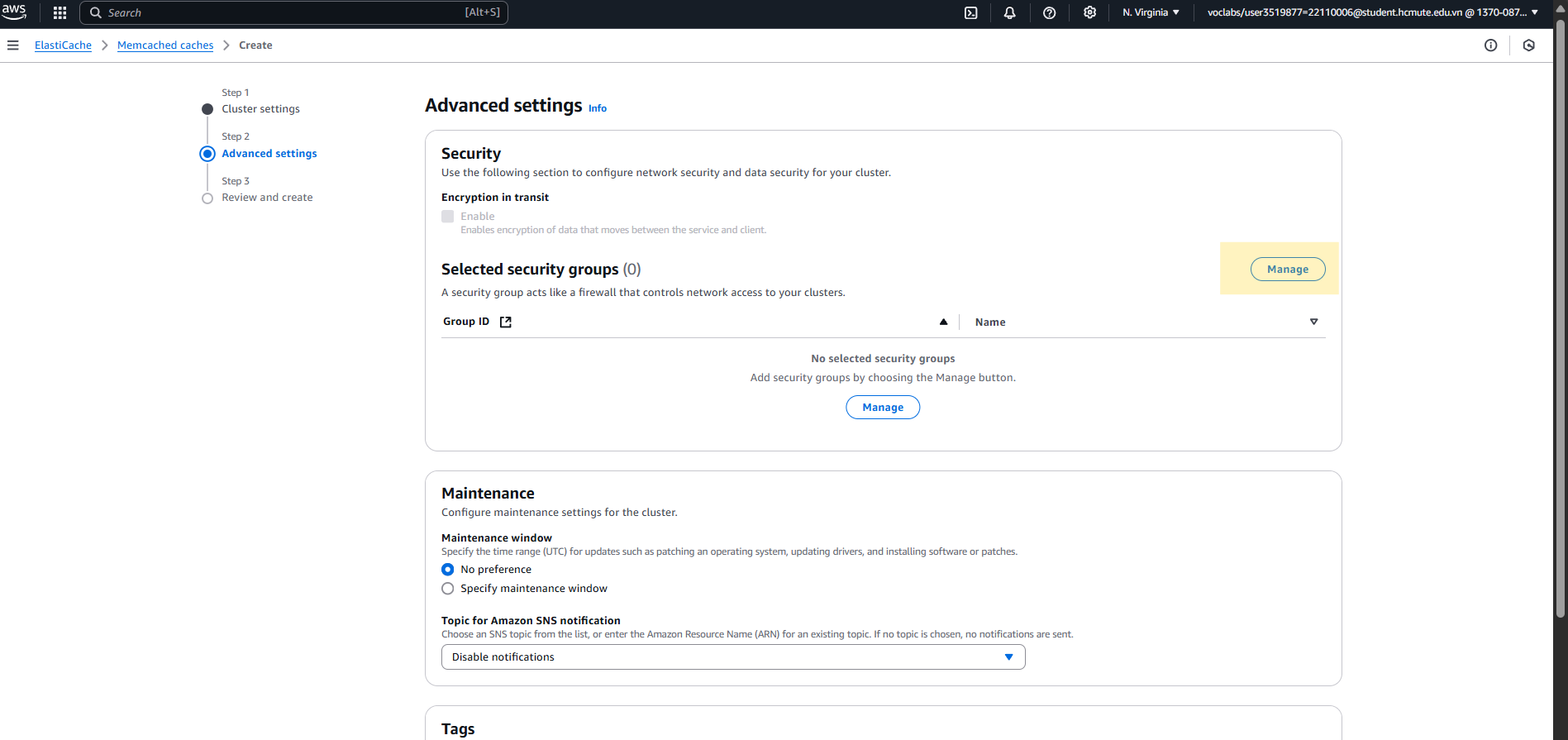
A screenshot of a computer

Description automatically generated

Bấm Next.

- Cài đặt **Security groups**

Bấm Manage:



Chọn security group có chữ Cluster trong tên:

A screenshot of a computer

Description automatically generated

Bấm Choose 🡪 Next.

Review lại các cài đặt, xong rồi bấm Create.

A screenshot of a computer

Description automatically generated

## **Task 4: Loading records into the cache from the database**

### **13. Thay đổi thư mục thực thi trên AWS Cloud9 terminal:**

Quay lại service Cloud9, mở terminal chạy lệnh:

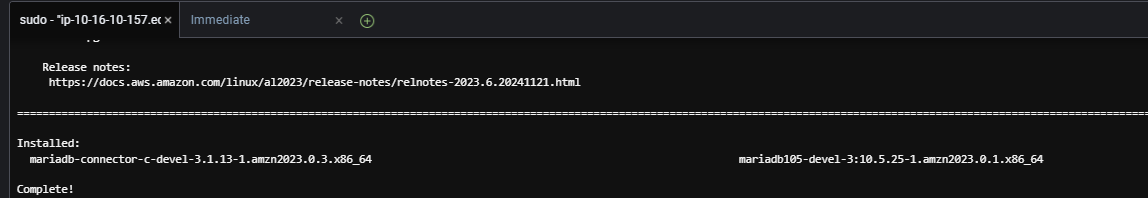
|  |
| --- |
| cd ~/environment/python\_3 |

A black rectangular object with a black background

Description automatically generated

### **14. Cài đặt các thư viện mà Python cần để tương tác với CSDL và cache:**

|  |
| --- |
| sudo dnf install -y mariadb105-devel gcc python3-devel |

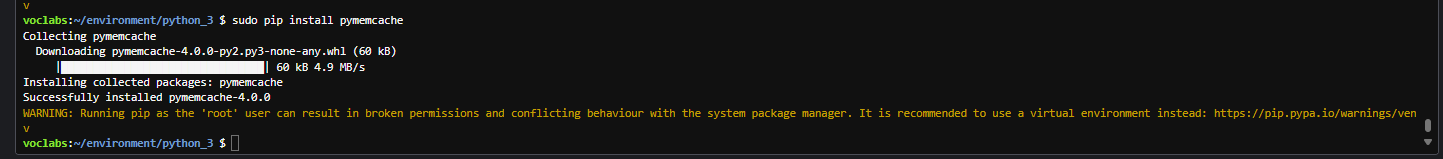


|  |
| --- |
| sudo pip install mysqlclient |

A black screen with white text

Description automatically generated

|  |
| --- |
| sudo pip install pymemcache |



### **15. Tìm endpoints ElastiCache cho Memcached.**

|  |
| --- |
| aws elasticache describe-cache-clusters |

A screenshot of a computer

Description automatically generated

Kết quả: ta lấy được Address của ConfigurationEndpoint như sau:

|  |
| --- |
| {  "CacheClusters": [  {  "CacheClusterId": "memcachedcache",  "ConfigurationEndpoint": {  "Address": "memcachedcache.bil4xc.cfg.use1.cache.amazonaws.com",  "Port": 11211  },  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",  "CacheNodeType": "cache.r6g.large",  "Engine": "memcached",  "EngineVersion": "1.6.6",  "CacheClusterStatus": "available",  "NumCacheNodes": 3,  "PreferredAvailabilityZone": "Multiple",  "CacheClusterCreateTime": "2024-12-08T16:13:25.963000+00:00",  "PreferredMaintenanceWindow": "sat:06:30-sat:07:30",  "PendingModifiedValues": {},  "CacheSecurityGroups": [],  "CacheParameterGroup": {  "CacheParameterGroupName": "default.memcached1.6",  "ParameterApplyStatus": "in-sync",  "CacheNodeIdsToReboot": []  },  "CacheSubnetGroupName": "elasticachesubnetgroup",  "AutoMinorVersionUpgrade": true,  "SecurityGroups": [  {  "SecurityGroupId": "sg-06f198058a81dbf78",  "Status": "active"  }  ],  "AuthTokenEnabled": false,  "TransitEncryptionEnabled": false,  "AtRestEncryptionEnabled": false,  "ARN": "arn:aws:elasticache:us-east-1:137008761826:cluster:memcachedcache",  "ReplicationGroupLogDeliveryEnabled": false,  "LogDeliveryConfigurations": [],  "NetworkType": "ipv4",  "IpDiscovery": "ipv4"  }  ]  } |

Gõ q để thoát.

### **16. Tìm Aurora Serverless cluster endpoint**

|  |
| --- |
| aws rds describe-db-cluster-endpoints |



Kết quả:

|  |
| --- |
| {  "DBClusterEndpoints": [  {  "DBClusterIdentifier": "supplierdb",  "Endpoint": "supplierdb.cluster-c988c0wyuor2.us-east-1.rds.amazonaws.com",  "Status": "available",  "EndpointType": "WRITER"  }  ]  } |

### **17. Kiểm tra lại script sẽ được sử dụng để kiểm tra việc tải dữ liệu vào cache:**

Mở file *python\_3* / *find\_all.py*

- Đầu tiên sẽ truy vấn bộ nhớ đệm Memcached:

|  |
| --- |
| data = memcached\_client.get('all\_beans') |

- Nếu thấy có dữ liệu trong Memcached, thì print ra console: Data returned from cache.

- Nếu không thấy có dữ liệu trong Memcached, sẽ truy vấn CSDL Aurora Serverless:

|  |
| --- |
| db\_query = "SELECT \* FROM beans"  mycursor = mydb.cursor()  mycursor.execute(db\_query) |

Và lưu lại vào Memcached:

|  |
| --- |
| memcached\_call = memcached\_client.set('all\_beans', output\_json, TTL\_INT) |

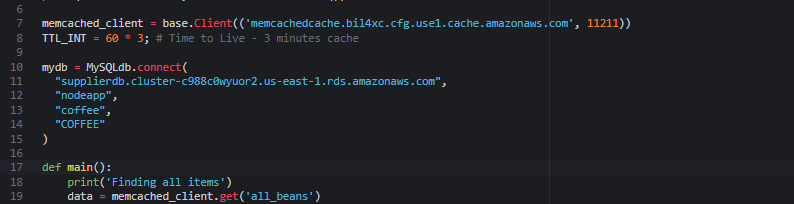
### **18. Cập nhật script để define kết nối CSDL và cache:**

- Cập nhật ElastiCache endpoint (lấy link từ bước 15)

|  |
| --- |
| memcached\_client = base.Client(('memcachedcache.bil4xc.cfg.use1.cache.amazonaws.com', 11211)) |

- Cập nhật Aurora endpoint (link lấy từ bước 16)

|  |
| --- |
| mydb = MySQLdb.connect(  "supplierdb.cluster-c988c0wyuor2.us-east-1.rds.amazonaws.com",  "nodeapp",  "coffee",  "COFFEE"  ) |



### **19. Chạy file python3 find\_all.py**

|  |
| --- |
| python3 find\_all.py |

A screenshot of a computer

Description automatically generated

Ta thấy data lấy từ CSDL, sau đó sẽ lưu lại vào cache.

Chạy lệnh trên lại 1 lần nữa:

A screenshot of a computer

Description automatically generated

Lần này dữ liệu được lấy trược tiếp từ cache.

Tuy nhiên, sau 3 phút (TTL = 180), thì dữ liệu trong cache sẽ bị xóa đi. Nên nếu chạy câu lệnh trên 1 lần nữa, thì sẽ ra kết quả như hình 1 (Setting result in cache).

## **Task 5: Updating data**

### **20. Kiểm tra script sẽ được sử dụng để kiểm tra bản cập nhật cho bản ghi trong cơ sở dữ liệu Aurora.**

- Mở file *update\_item.py và review code.*

### **21. Cập nhật script update\_item.py:**

Cập nhật các endpoints như bước 18.

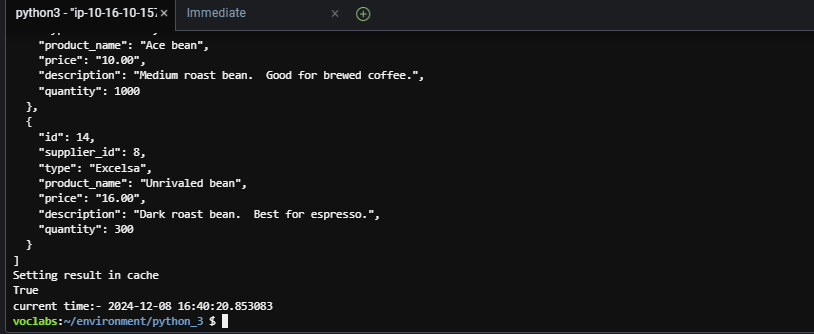
|  |
| --- |
| memcached\_client = base.Client(('memcachedcache.bil4xc.cfg.use1.cache.amazonaws.com', 11211))  TTL\_INT = 60 \* 3; # 3 min cache  mydb = MySQLdb.connect(  "supplierdb.cluster-c988c0wyuor2.us-east-1.rds.amazonaws.com",  "nodeapp",  "coffee",  "COFFEE"  ) |

Sau đó, chạy file:

|  |
| --- |
| python3 update\_item.py |

### **22. Kiểm tra việc làm mới bộ đệm.**

|  |
| --- |
| python3 find\_all.py |



Chạy lại lần nữa để đảm bảo dữ liệu lấy từ cached:

A screenshot of a computer

Description automatically generated

## **Task 6: Creating a new record**

### **23. Kiểm tra script sẽ được sử dụng để kiểm tra việc tạo bản ghi trong cơ sở dữ liệu Aurora.**

Mở file *create\_item.py:*

### **24. Chạy file create\_item.py:**

- Cập nhật các endpoints (như bước 18)

|  |
| --- |
| memcached\_client = base.Client(('memcachedcache.bil4xc.cfg.use1.cache.amazonaws.com', 11211))  TTL\_INT = 60 \* 3; # 3 min cache  mydb = MySQLdb.connect(  "supplierdb.cluster-c988c0wyuor2.us-east-1.rds.amazonaws.com",  "nodeapp",  "coffee",  "COFFEE"  ) |

- Chạy file:

|  |
| --- |
| python3 create\_item.py |

A screenshot of a computer

Description automatically generated

### **25. Kiểm tra lại việc làm mới bộ đệm:**

|  |
| --- |
| python3 find\_all.py |

A screen shot of a computer

Description automatically generated

Chạy lại lần nữa để thấy data được lấy từ cache:

A computer screen shot of a black screen

Description automatically generated

## **Task 7: Deleting a record**

### **26. Xem lại script sẽ được sử dụng để kiểm tra việc xóa bản ghi khỏi cơ sở dữ liệu Aurora.**

Mở file *delete\_item.py.*

### **27. Chạy file delete\_item.py**

- Update các endpoints (như bước 18)

|  |
| --- |
| memcached\_client = base.Client(('memcachedcache.bil4xc.cfg.use1.cache.amazonaws.com', 11211))  TTL\_INT = 60 \* 3; # 3 min cache  mydb = MySQLdb.connect(  "supplierdb.cluster-c988c0wyuor2.us-east-1.rds.amazonaws.com",  "nodeapp",  "coffee",  "COFFEE"  ) |

- Chạy file python3 delete\_item.py

|  |
| --- |
| python3 delete\_item.py |

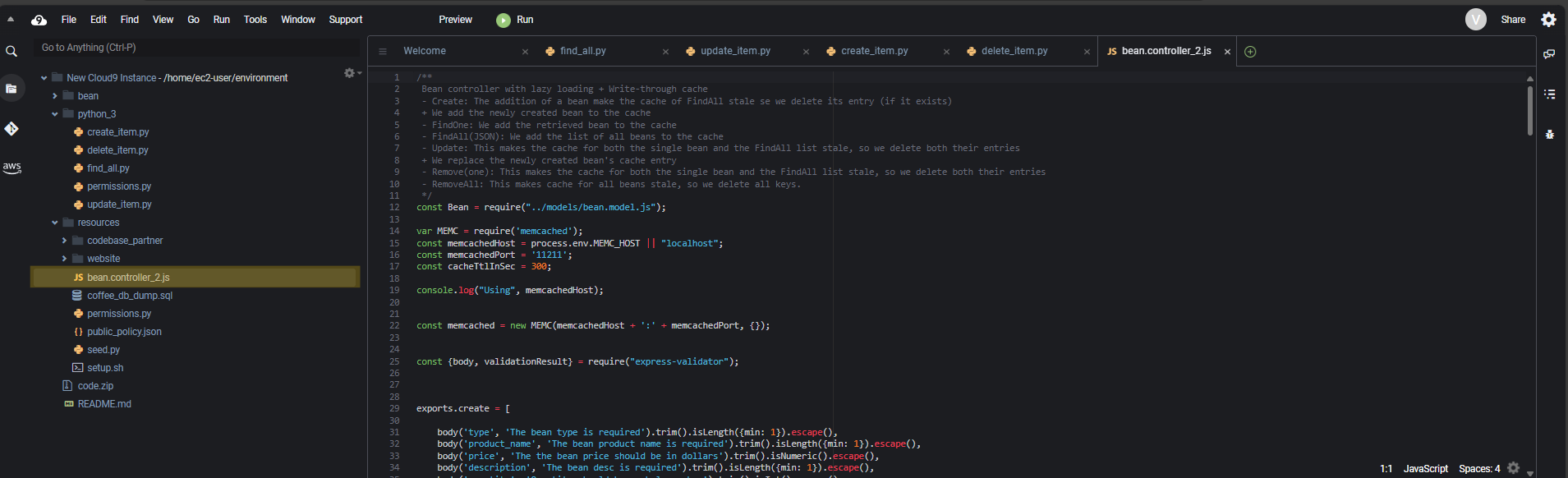
A screen shot of a computer

Description automatically generated

## **Task 8: Updating the application container**

### **28. Xem lại code mới dùng Node.js**

Mở file resources / *bean.controller\_2.js*



### **29. Cập nhật code ứng dụng Node.js**

|  |
| --- |
| mv ~/environment/resources/bean.controller\_2.js \  ~/environment/resources/codebase\_partner/app/controller/bean.controller.js |

A screenshot of a computer program

Description automatically generated

### **30. Rebuild và cập nhật Docker image**

|  |
| --- |
| cd ~/environment/resources/codebase\_partner/  npm install  docker build --tag node\_app . |

A black rectangular object with a black border

Description automatically generated

### **31. Cho phép ứng dụng Docker client kết nối với dịch vụ Amazon Elastic Container Register (Amazon ECR).**

- Mở AWS Management Console, tìm tên người dùng (bắt đầu với voclabs/user)

A screenshot of a computer screen

Description automatically generated

Account ID: 137008761826

- Mở AWS Cloud9 terminal: (thay Account ID tương ứng với tài khoản của mình)

|  |
| --- |
| aws ecr get-login-password \  --region us-east-1 | docker login --username AWS \  --password-stdin 137008761826.dkr.ecr.us-east-1.amazonaws.com |

A screen shot of a computer

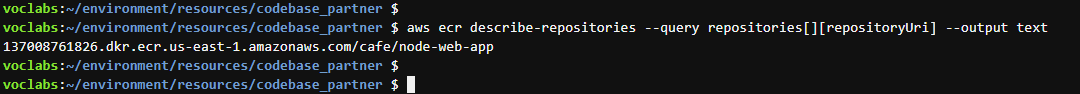
Description automatically generated

### **32. Push container image đã cập nhật lên Amazon ECR:**

- Tìm URI của repo:

|  |
| --- |
| aws ecr describe-repositories --query repositories[][repositoryUri] --output text |

Kết quả: 137008761826.dkr.ecr.us-east-1.amazonaws.com/cafe/node-web-app



- Gắn tag cho Amazon ECR image:

|  |
| --- |
| docker tag node\_app:latest 137008761826.dkr.ecr.us-east-1.amazonaws.com/cafe/node-web-app:latest |

- Push the image to Amazon ECR:

|  |
| --- |
| docker push 137008761826.dkr.ecr.us-east-1.amazonaws.com/cafe/node-web-app:latest |

A black screen with white text

Description automatically generated

### **33. Test your Elastic Beanstalk website.**

Bấm icon Cloud9 🡪 Go to your dashboard

A screenshot of a computer

Description automatically generated

Chọn **Services** 🡪 **Elastic Beanstalk**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Chọn MyEnv 🡪 click vào link Domain:

<http://myenv.eba-y2t7hzyg.us-east-1.elasticbeanstalk.com/>

A screenshot of a computer

Description automatically generated

Kết quả:

A screen shot of a computer

Description automatically generated

Thêm /beans vào cuối URL:

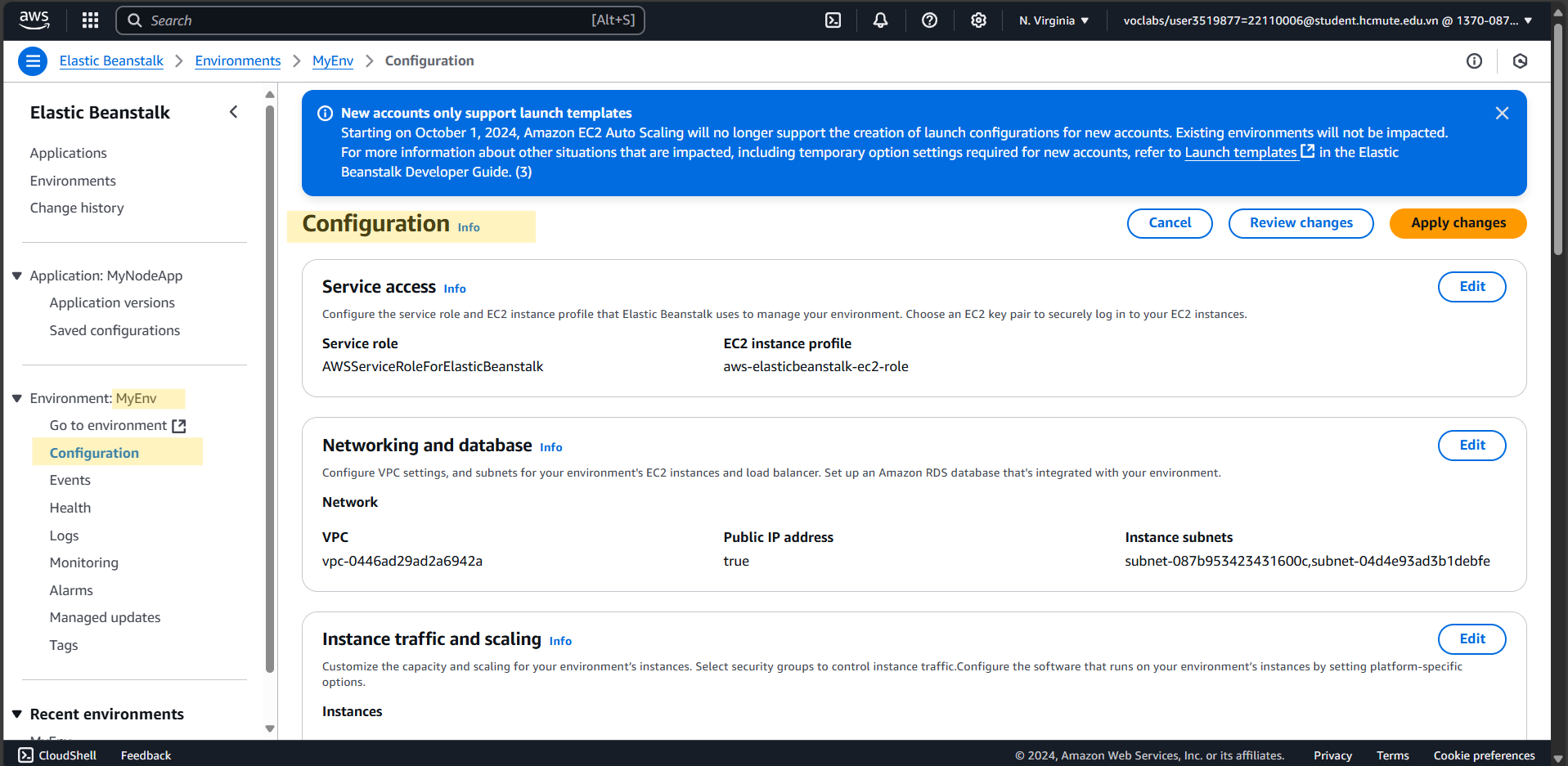
<https://myenv.eba-y2t7hzyg.us-east-1.elasticbeanstalk.com/beans>

A screenshot of a computer

Description automatically generated

### **34. Cấu hình biến môi trường Memcached mới, MEMC\_HOST.**

Quay lại Elastic Beanstalk 🡪 Chọn Configuration cho MyEnv:



Kéo xuống phần **Updates, monitoring, and logging 🡪 chọn Edit:**

**A screenshot of a computer

Description automatically generated**

Thêm biến môi trường như hình:

* **Name:** MEMC\_HOST
* **Value:** Endpoint Memcached lấy ở bước 18.

A screenshot of a computer

Description automatically generated

Bấm Apply

Quay lại website, xem /beans

<http://myenv.eba-y2t7hzyg.us-east-1.elasticbeanstalk.com/beans>

A screenshot of a computer

Description automatically generated

### **35. Kiểm tra code cập nhật từ trang web quán cà phê.**

- Mở Service **S3 🡪** chọn bucket có chứa **s3bucket** trong tên:

A screenshot of a computer

Description automatically generated

Chọn tab **Permissions** 🡪 **Block public access (bucket settings) 🡪 Chọn Edit:**

A screenshot of a computer

Description automatically generated

Uncheck hết các options 🡪 Save changes:

A screenshot of a computer

Description automatically generated

Gõ confirm để xác nhận:

A screenshot of a computer

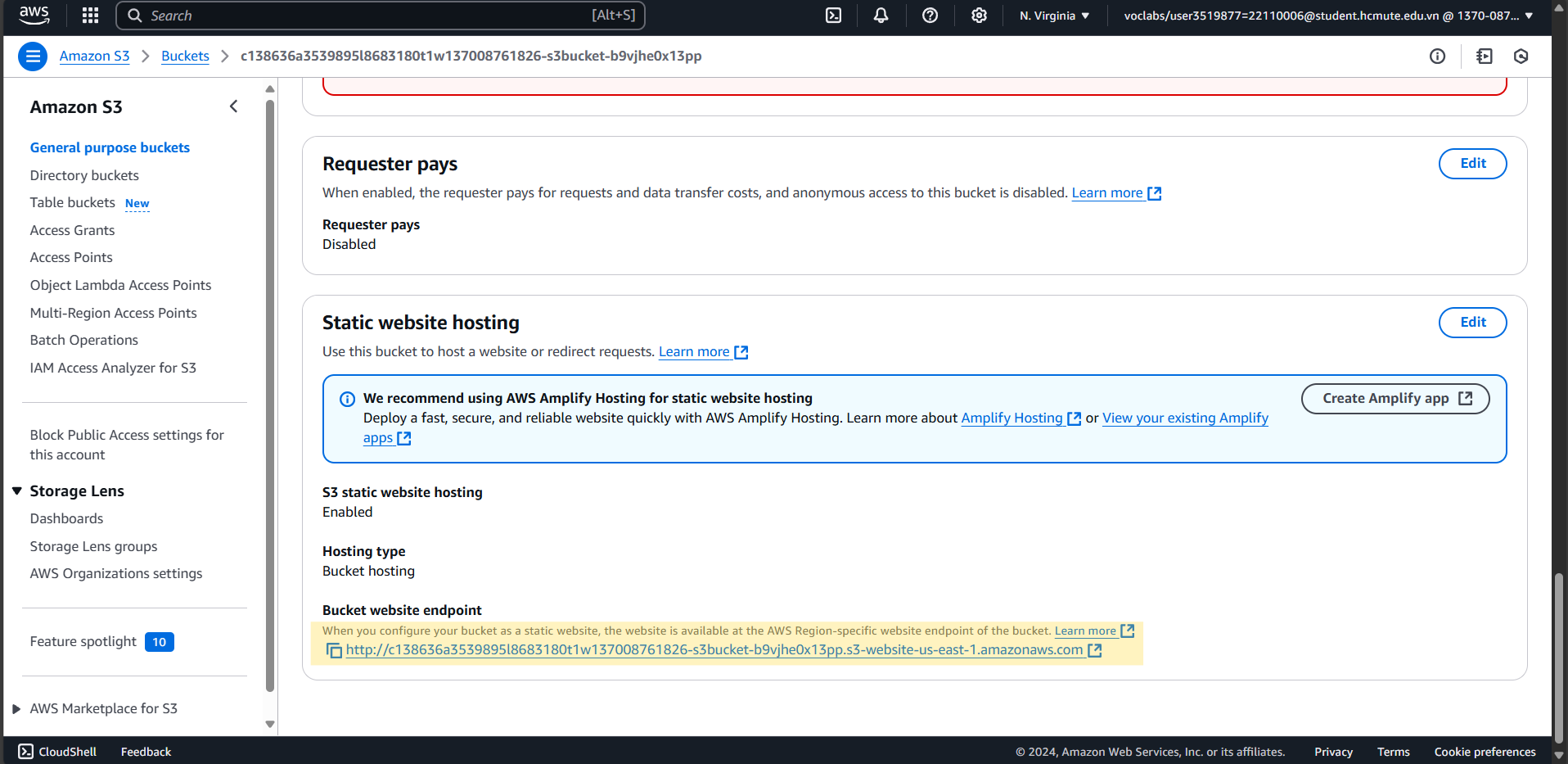
Description automatically generated

Quay lại chọn tab Properties:

A screenshot of a computer

Description automatically generated

Kéo xuống phần **Static website hosting,** mở URL trong tab mới.



Xong. Quay về Submit.

