|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Declaration** |  | | | | |
| Questions in this exercise are intentionally complex and could be convoluted or confusing. This is by design and to simulate real life situations where customers seldom give crystal clear requirements and ask unambiguous questions. | | | | | |
|
|
| I have read the above statement and agree to these conditions | | | | | |
| I AGREE | Jyoti Mohanty | | | | |
| <Enter your name above this line to indicate that you are in agreement> | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Instructions** |  |  |  |  |  |
| Every screenshot requested in this workbook is compulsory and carries 1 marks | | | | | |
| Your AWS account ID must be clearly visible in every screenshot using the AWS console; missing id or using someone else's id is not permitted. Such cases will be considered as plagiarism and severe penalty will be imposed. | | | | | |
| All screenshots must be in the order mentioned under "Expected Screenshots" for every step | | | | | |
| DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances. | | | | | |
| The file should be renamed in the format BATCH\_FIRSTNAME\_LASTNAME\_PROJECT1.  For example: PGPCCMAY18\_VIJAY\_DWIVEDI\_PROJECT1.pdf | | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Resource Clean Up** | |  |  |  |  |
| Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you’ve completed your lab or project is always necessary. This is true whether you’re doing a lab or implementing a project at your workplace. | | | | | |
| After completing the lab, make sure to delete each resource created in reverse chronological order. | | | | | |

**Architecture diagram**

****

|  |  |
| --- | --- |
| **Architecture Implementation** | |
| 1 | Create and configure a 3 node Cassandra cluster on EC2 |
| 2 | Insert some sample data into the cluster |
| 3 | Check and change the consisterncy level of the cluster |

**STEP 1: Cassandra Cluster Creation and Configuration**

|  |  |
| --- | --- |
| Step number | A |
| Step name | Creation of Cluster |
| Instructions | 1) Create a new security group by the name "cassandra-ports"  - "Custom TCP"and port range of "1024-65500" and port 22 for SSH  2) Create 3 EC2 instances (t2-small) using the 7 step workflow  - Use the "SKL-Ubuntu-Cassandra" (community AMI) in AZ1, AZ2, AZ3  - The AMI contains - Ubuntu 16.04, JDK8, Python2.7 & unconfigured Cassandra  - Assign the "cassandra-ports" SG  - Make sure to name the instances as instance 1, 2, and 3 to make it easier to keep track of them  3) Follow the steps to configure Cassandra (repeat this in all instances) by logging in via SSH into each of the instances  - Go to the "conf" folder and update the cassandra.yaml using the commands below  *- cd /opt/apache-cassandra-3.11.1/conf*  - *sed -i 's=MOD\_CLUSTER\_NAME=GL-Cluster=g' cassandra.yaml*  *- sed -i 's=MOD\_IP\_ADDRESS=[LOCAL IP here]=g' cassandra.yaml*  *- sed -i 's=MOD\_SEED\_LIST=[SEED IP here]=g' cassandra.yaml*  *- sed -i 's=MOD\_DATACENTER=dc1=g' cassandra-rackdc.properties*  *- sed -i 's=MOD\_RACK=[RACK 1,2,3]=g' cassandra-rackdc.properties (r1 for instance 1, r2 for instance 2 and r3 for instance 3)*  Example commands for the above are shown below (THESE COMMANDS ARE JUST AN EXAMPLE AND WILL NOT WORK IN YOUR SETUP)  *sed -i 's=MOD\_IP\_ADDRESS=172.31.39.217=g' cassandra.yaml*  *sed -i 's=MOD\_CLUSTER\_NAME=GL-Cluster=g' cassandra.yaml*  *sed -i 's=MOD\_SEED\_LIST=172.31.43.230=g' cassandra.yaml*  *sed -i 's=MOD\_DATACENTER=dc1=g' cassandra-rackdc.properties*  *sed -i 's=MOD\_RACK=r2=g' cassandra-rackdc.properties*  *- Start the Cassandra Daemon in each instance using the commands below*  *- cd /opt/apache-cassandra-3.11.1/bin*  *- ./cassandra*  *- ./nodetool status*  Expected Screenshots : 1) EC2 screen showing all 3 instances after creations  2) Cassandra configuration commands on Instance 2  3) Nodetool status output for instance 1 |
|  |  |

**<Insert screenshot for a(1) here>**

**ˇ** **Graphical user interface, text, application, email

Description automatically generated**

**<Insert screenshot for a(2) here>**

**A picture containing diagram

Description automatically generated**

**A computer screen capture

Description automatically generated with low confidence**

**<Insert screenshot for a(3) here>**

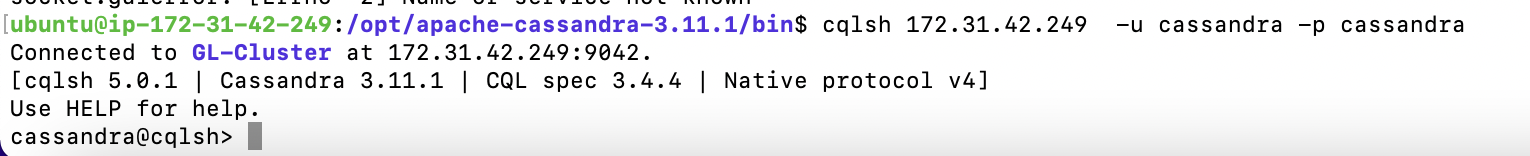
**Text, letter

Description automatically generated**

**STEP 2 : Uploading of Sample Data**

|  |  |  |
| --- | --- | --- |
| Step number | a) |  |
| Step name | Insert Sample Data into Cluster |  |
| Instructions | 1) Login to cqlsh in instance 1 by logging into it via SSH and running the below command  *cd /opt/apache-cassandra-3.11.1/bin*  *./cqlsh <EC2 private IP> -u cassandra -p cassandra*  2) Ensure the "system\_auth" keyspace replication is changed using the command below  *ALTER KEYSPACE "system\_auth" WITH REPLICATION = {'class':'NetworkTopologyStrategy', 'dc1':3};*  3) Create a keyspace and sample table using the command below  *CREATE KEYSPACE IF NOT EXISTS starfleet WITH replication = {'class':'NetworkTopologyStrategy', 'dc1':3};*  *CREATE TABLE starfleet.user (*  *user\_id VARCHAR,*  *location VARCHAR,*  *display\_name VARCHAR,*  *first\_name VARCHAR,*  *last\_name VARCHAR,*  *PRIMARY KEY (user\_id, location)*  *)*;  4) Insert 2 records into the table using the commands below  *INSERT INTO starfleet.user (user\_id,location,display\_name,first\_name,last\_name)*  *VALUES ('u1','earth1','Kirk','William','Shatner');*  *INSERT INTO starfleet.user (user\_id,location,display\_name,first\_name,last\_name)*  *VALUES ('u2','vulcan','Spock','Leonard','Nimoy');*  Required Screenshots : 1) Logging into cqlsh  2) Creation of user table  3) Inserting of the data records |  |

**<Insert screenshot for a(1) here>**

****

**<Insert screenshot for a(2) here>**

**Graphical user interface, text, application

Description automatically generated**

**<Insert screenshot for a(3) here>**

**Graphical user interface, text

Description automatically generated**

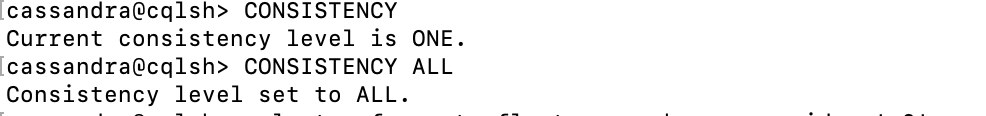
**STEP 3 : Checking consistency levels**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step number | a |  |  |  |
| Step name | Check and modify the consistency levels of the cluster |  |  |  |
| Instructions | 1) Login to cqlsh of instance 1 as done previously  2) Check the consistency level of the cluster with the command below  *CONSISTENCY*  3) Set the consistency level to ALL with the command below  CONSISTENCY ALL  4) Run the following SQL command  select \* from starfleet.user where user\_id = 'u2';  5) Shut down node 3 with by logging in via SSH to instance 3 and running the command below  *cd /opt/apache-cassandra-3.11.1/bin*  *./nodetool stopdaemon*  6) Go back to the cqlsh shell of instance 1 and run the above SQL command again  7) Set the consistency level to QUORUM and run the SQL command again | | | |
| Expected screenshots | 1) Initial Consistency Level 2) Setting Consistency level to ALL 3) Output of SQL command with consistency level all after shutting down node 3  4) Setting consistency level to QUORUM and SQL output | | | |

**<Insert screenshot for a(1) here>**

****

**<Insert screenshot for a(2) here>**

****

**<Insert screenshot for a(3) here>**

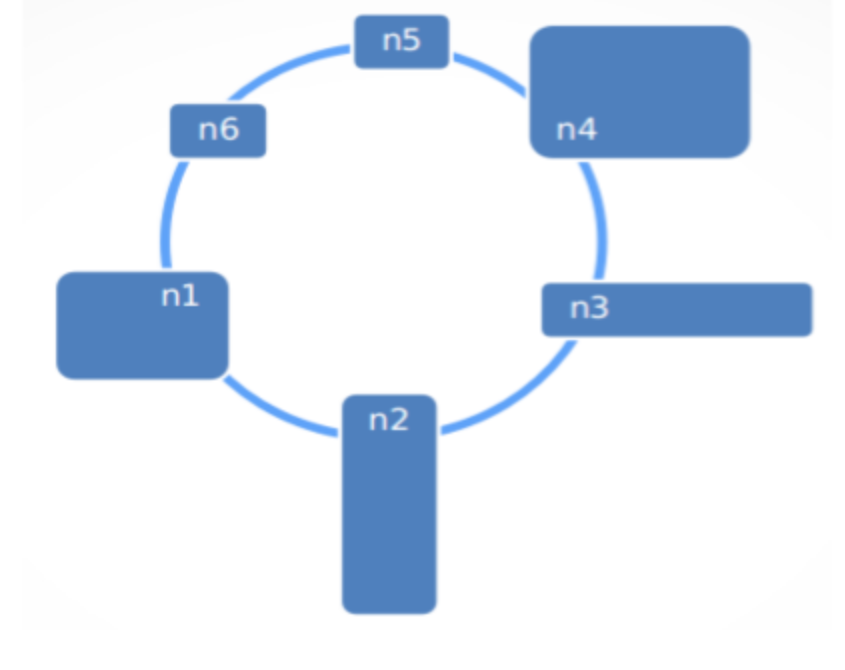
****

**<Insert screenshot for a(4) here>**

**A picture containing graphical user interface

Description automatically generated**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Answer the following questions** | | | | **Marks** |
| Q1 | How many racks are the 3 nodes created in this exercise distributed among | | | 1 |
|  | a) 1 |  |  |  |
|  | b) 3 |  |  |  |
|  | c) Variable and left to the learner to decide |  |  |  |
|  | d) 2 |  |  |  |
|  | Enter your answer here | B |  |  |
|  |  |  |  |  |
| Q2 | What is the replication factor of the cluster in this exercise? | | | 1 |
|  | a) 3 |  |  |  |
|  | b) Not defined |  |  |  |
|  | c) Different for each node |  |  |  |
|  | d) 1 |  |  |  |
|  | Enter your answer here | A |  |  |
|  |  |  |  |  |
| Q3 | How many replicas must respond for the query to be successful in consistency level Quorum in this exercise? | | | 1 |
|  | a) Exactly 2 |  |  |  |
|  | b) Atleast 2 |  |  |  |
|  | c) Cannot be determined |  |  |  |
|  | d) More than 1 |  |  |  |
|  | Enter your answer here | D |  |  |
|  |  |  |  |  |
| Q4 | Why did the query with consistency level ALL fail in this exercise? | | | 1 |
|  | a) Live replicas < Replication factor |  |  |  |
|  | b) Number of responses < Number of nodes |  |  |  |
|  | c) Data was not saved on node 3 |  |  |  |
|  | d) Not enough information to specify a reason |  |  |  |
|  | Enter your answer here | b |  |  |
|  |  |  |  |  |
| Q5 | How can the partitioner used in a cluster be changed with no impact? | | | 3 |
|  |  |
|  | Create a new cluster that’s murmur3  Write to both clusters  In background read and copy data to new cluster while writes are duplicated  Once background job is complete flips reads from old cluster to new and decommission old cluster | | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |
| Q6 | Look at the picture below and argue for or against such a topology. Assume y-axis is memory and x-axis is cpu. So n3 has more cpu than memory. | | | 4 |
|  |  |  |  |  |
|  |  |  |  |  |
|  | Because CPU is more as compared d to memory | | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  | **Max marks** | **11** |

****

|  |  |
| --- | --- |
| **Grades distribution** |  |
| MCQs | 4 (1 mark each) |
| Subjective questions | 7 marks (3+4) |
| Implementation screenshots | 9 marks (1 marks each) |
| Total | 20 marks |