Following infrastructure was created on aws to complete the assignment:

1. 3 x t3a.medium (2cpu, 4GB ram, 100GB SSD gp2) - For running YB-Master and YB-TServer
2. 1 x t2.micro ( 1cpu, 1GB ram, 8GB SSD gp2) – For running load generation application

All the tasks were completed except SQLInsert, and codebase has been pushed to my public GitHub repository for future updates and additional learning that I wish to continue post assignment. <https://github.com/shantanugupta/yugabyte-learning>

I created following files to make deployment process convenient for myself. This development has been broken into following files

1. [configfile.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/configfile.bash) – All the environment variables were exported into this file including variables listed in [official documentation](https://docs.yugabyte.com/latest/deploy/public-clouds/aws/manual-deployment/#running-sample-workload).
2. [setup\_machine.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/setup_machine.bash) – This file was used to load configfile.bash and run through manual deployment steps listed in [official documentation](https://docs.yugabyte.com/latest/deploy/public-clouds/aws/manual-deployment/#running-sample-workload). Following tasks were performed to setup yugabyte universe
   1. Load environment variables
   2. Prepare/Configure/Mount data drives
   3. Install required & optional packages e.g., ntp
   4. Set uLimits
   5. Install Yugabyte DB
   6. Setup softlinks
   7. Configure master, tserver
   8. Start YB-Master using [start\_master.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/start_master.bash)
   9. Start YB-TServer using [start\_tserver.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/start_tserver.bash)
   10. Configure region aware placement - Optional (Did a mistake in configuring causing timeout when creating table)
   11. Set zone preference (Optional) – I did it since I created two nodes in AZ-2 and one in AZ-1
   12. Test ybsql using [test-pgsql.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/test-pgsql.bash)
   13. Test ybcsql using [test-cassandra.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/test-cassandra.bash)
   14. Configured prometheus using [yugabytedb.yml](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/yugabytedb.yml)
3. [setup.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/setup.bash) – This was used as a utility file to copy files between different nodes. Setting up cronjob. Installation of optional packages for debugging or testing.
4. [start\_master.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/start_master.bash) – Script to start YB-Master on all of the nodes in a cluster
5. [start\_tserver.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/start_tserver.bash) – Script to start YB-TServer on all of the nodes in a cluster
6. [stopNdel.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/stopNdel.bash) – Script to stop YB-Master and YB-TServer services on all the nodes in a cluster and delete data directory containing data.
7. [yugabytedb.yml](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/yugabytedb.yml) – Prometheus configuration related settings
8. [test-cassandra.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/test-cassandra.bash) – Cassandra scripts to run on ybcql shell to test basic querying operations
9. [test-pgsql.bash](https://github.com/shantanugupta/yugabyte-learning/blob/main/yb/test-pgsql.bash) – SQL script to run on ybsql shell to test and load Northwind database on the cluster. This file uses ~/yb/northwind\_ddl.sql & ~/yb/northwind\_data.sql files for loading northwind database.

Experience/Issues identified and learned:

1. I mistakenly configured “region aware placement” incorrectly and got an issue when testing sql scripts in ybsql.
   1. I was able to login to ybsql and create database
   2. When trying to create table, I got error message shown was “request timeout”. Upon checking tserver logs at /mnt/d0/yb-data/tserver/logs/yb-tserver.ERROR, I noticed error stating No of nodes required in placement group is 3 while found 2 based on which I was able to resolve the issue.
2. When running CassandraBatchTimeseries load on server, one of the tserver when down i.e., DEAD, upon checking the logs, I identified the issue was with configured clock skew time of 0.5 seconds was reached and node was out of clock sync so was taken out. I restarted tserver service to bring it back and replication started again.

Overall experience:

Documentation was super awesome. Deployment process was flawless.

CPU utilization metrics were captured by cronjob. Below listed table and graph shows CPU utilization for CassandraKeyValue application log as well as CassandraBatchTimeSeries load.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CPU Utilization (Avg)** | | | | | | | | |
| **Time** | **%User** | | | **%System** | | | **Total %user** | **Total %system** |
| **10-0-0-66** | **10-0-1-108** | **10-0-1-155** | **10-0-0-66** | **10-0-1-108** | **10-0-1-155** |
| 5 PM | 1.20 | 1.61 | 7.28 | 1.20 | 0.91 | 5.22 | 1.99 | 1.65 |
| 6 PM | 6.73 | 14.68 | 21.51 | 5.19 | 7.45 | 10.57 | 14.31 | 7.74 |
| 7 PM | 54.38 | 93.10 | 93.32 | 3.23 | 4.57 | 4.36 | 80.27 | 4.05 |
| 8 PM | 54.24 | 88.53 | 90.36 | 3.04 | 4.29 | 4.65 | 77.70 | 3.99 |
| 9 PM | 55.08 | 84.67 | 86.85 | 2.87 | 4.00 | 4.57 | 75.53 | 3.81 |
| 10 PM | 50.12 | 69.78 | 86.40 | 2.67 | 3.33 | 4.09 | 68.76 | 3.36 |
| 11 PM | 11.47 | 0.06 | 62.94 | 1.08 | 0.03 | 2.71 | 24.81 | 1.28 |
| Total | 33.34 | 56.98 | 71.84 | 2.76 | 3.86 | 5.16 | **53.16** | **3.88** |



**Graphs of running CassandraKeyValue load:**

Approx 1GB data was generated in this process and this process ran for 1hr approx.

**Graphs of running CassandraBatchTimeseries load:**

This process was executed for approx. 8 hours and it generated approx. 19 GB data. During an execution of this app, clock sync related issues was observed thrice and one of the TServer node went down during that process. TServer service was restarted to keep all the nodes in running state.