Account
Dashboard
Courses
Groups
Calendar
Inbox
Arc
Help

Spring 2019
Home
Announcements
Assignments
Grades
People
Pages
Quizzes

CMPE 281 - Personal NoSQL Project

Due Mar 30 by 11:59pm **Points** 100 **Submitting** a text entry box
Available until Mar 30 at 11:59pm

This assignment was locked Mar 30 at 11:59pm.

Objectives:

In this project, you will be testing the **Partition Tolerance** of two NoSQL DBs using the procedures described in the following article: https://www.infoq.com/articles/jepsen . Note: you don't have to follow the directions in the article "verbatim". Adjust the steps as you see necessary -- for example, in how you create a partition in AWS.

In addition, you will be developing an **Data Service API in Go** on top of your Database Setup and deploying the API on AWS for the Team portion later in the course.

Notes:

- Use the steps in the article as guidance only.
- Feel free to diverge from those steps as needed to accomplish your testing goals.
- Using alternative testing programs, programming languages, tools and/or approaches to creating network partitions are allowed.
- Please document any steps you take that diverge from the steps in the article.

Requirements:

- Select **one CP and one AP** NoSQL Database. For example: **Mongo** and **Riak.**
- **Note: Other NoSQL DBs that can be configured in CP or AP mode are also allowed.**
- **For each Database:**
  - Set up your cluster as **AWS EC2 Instances. (# of Nodes and Topology is open per your design)**
    - **Make sure to note your approach to creating a "network partition" for experiments.**
  - Set up the **Experiments** (i.e. Test Cases) to answer the following questions:
    - **CP:**
      - **How does the system function during normal mode (i.e. no partition)**
      - **What happens to the master node during a partition?**
      - **Can stale data be read from a slave node during a partition?**
      - **What happens to the system during partition recovery?**
    - **AP:**
      - **How does the system function during normal mode (i.e. no partition)**
      - **What happens to the nodes during a partition?**
      - **Can stale data be read from a node during a partition?**
      - **What happens to the system during partition recovery?**
  - Run the **Experiments** and **Record results.**
- Project must use a **GitHub Repo** (assigned by TA) to maintain source code, documentation and design diagrams.
  - Repo will be maintain in: https://github.com/nguyensjsu
- Keep a **Project Journal** (as a markdown document) recording weekly progress, challenges, tests and test results. ***Note: Keep this as detailed as needed for someone else reading your notes to be able to reproduce your experiments and results.***
- All documentation must be written in GitHub Markdown (including diagrams) and committed to GitHub
  - https://help.github.com/articles/about-writing-and-formatting-on-github/
- ***Note: For the Team Hackathon Project, you may use \your NoSQL database(s) in this Project and develop the*** **Go API and integrate the API with the Team SaaS App**.

Grading

- Record a YouTube Video Demonstrations to include (***max 15 minutes***):
  - **Demonstration of AWS setup** of NoSQL DB - ***25 points***
    - *Just show config settings in AWS. This should take no more than a couple of minutes.*
    - *You don't have to show every step taken to bring up your cluster (these details should be in your Journal)*
  - **Demonstration of CP & AP properties** from Experiments showing Test Results - ***25 points***
    - *Show the output of querying a document during normal state*
    - *Create a Partition in your Cluster and then update the document*
    - *Show the output of querying the modified document from different nodes*
    - *Explain the results observed during partition mode*
    - *Restore the network (i.e. trigger a partition recovery in your cluster)*
    - *Show the output of querying the modified document from different nodes*

Submission

✓ **Submitted!**
Mar 30 at 10:50pm
Submission Details

Comments:
No Comments

        - *Explain the results observed post partition recovery.*
    - ***Guidelines for Video Recording.*** Please make sure that the video recordings for your personal project is no more than 15 minutes long. Focus on getting to the point and showing the following:
        1. Include a quick walk through on the AWS configurations (EC2 Instances, Security Groups, Etc..). This part should not take more than a minute long.
        2. Include a demo of your Test Scenario. I.E. Healthy Test, Create Partition, Unhealthy Test, Recovery, Back to Healthy Check. This should be the majority of the Video.
- Project Journal Entries:
    - **Project Journal** content recording incremental progress on setup, experiments and tests - ***50 points***
- Deductions will be based on:
    - **Frequency and Quality of commits** to the project Github.
    - **Quality** = is the Journal detailed enough to be "repeatable" by someone else?
    - This includes, but not limited to: **code, documentation and diagrams.**
    - As such, it is expected that all contributions must be visible via Github. See the following guidelines for how GitHub counts contributions: **https://help.github.com/articles/why-are-my-contributions-not-showing-up-on-my-profile/**

Also see: Comparison_of_Distribution_Technologies_in_Different_NoSQL_Database_Systems-SA_Dominik_Bruhn.pdf

Extra Credit (Max 20 Points)

A maximum of 20 points in extra credit related to this project can be earned using a combination of the following options. If more than 20 points are earned, only 20 points will be counted.

- **MongoDB Sharding (10 points)**
    - Configure your MongoDB Cluster to support "**Two Data Shards**".
    - Use the **Mongo Bios Collection** (from previous Labs) and decide on a "**Shard Key**".
- **Bonus Points for "Wow Factor" (10 points)**
    - I.E. Configurations and/or use of Features Beyond Materials in Class.

GitHub Commits

If you have multiple GitHub accounts (i.e. for SJSU and for Work), please make sure the following is configured for GitHub before committing to to SJSU GitHub Repos:

git config --global user.email "**$github_email**"
git config --global user.name **"$github_username**"

Where:

- **github_email** is your SJSU Email
- **github_username** is the GitHub Username associated with your SJSU Repo

Not doing so risks commits being excluded from **GitHub Insights Contributor Report.**

Submission:

- **Link to your GitHub Repo**
- **Link to your Your YouTube Video. (Max 15 Minutes)**
    - ***Demonstration of AWS setup** of NoSQL DB (with at least 5 nodes) - **25 points***
    - ***Demonstration of CP & AP properties** from Experiments showing Test Results - **25 points***
- **Link to your Project Journal on GitHub**
- ***NOTE: Your Personal project Repo will be transferred to your public GitHub account at the end of class.***