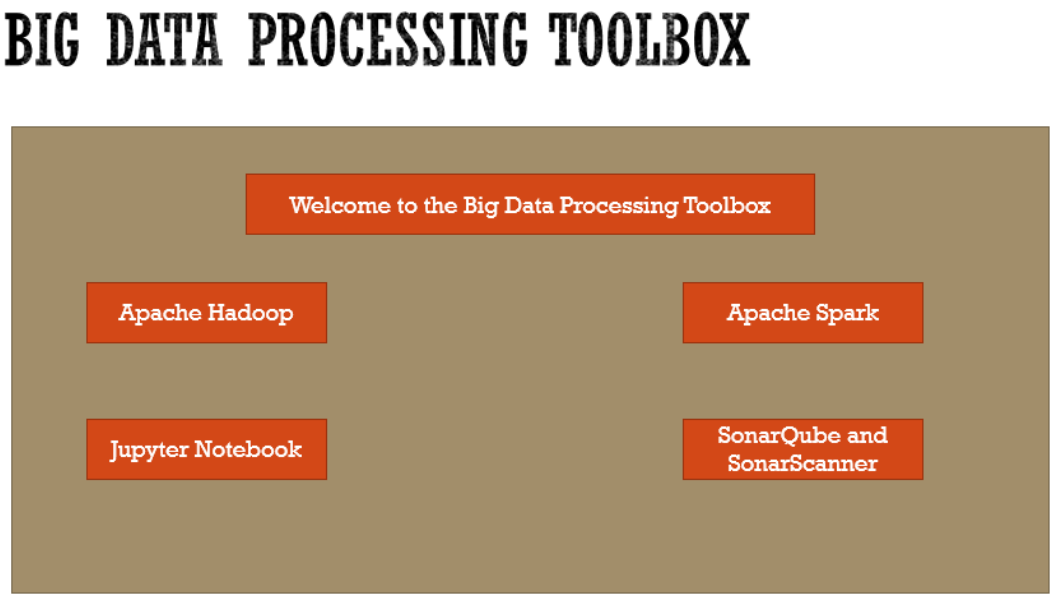
[Fall-2021]

Course Project – Option 1

Deadline: November 21st 11:59PM EST (8:59PM PST)

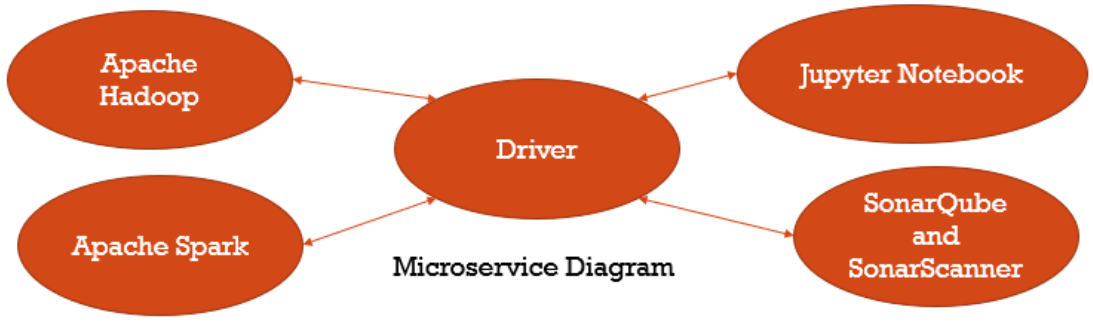


General Description:

You are requested to build a microservice-based application that would allow the users to run Apache Hadoop, Spark, Jupyter Notebooks, SonarQube and SonarScanner without having to install any of them.

Your application is supposed to have one main microservice that acts as the entry point for your overall application.

The expected architecture for your application would look like the following diagram:



- Each circle that is shown in the architecture diagram represents a **microservice that is hosted on a separate docker container**.
- **ALL** your docker containers should be deployed to **Kubernetes cluster**.
- Your **Kubernetes cluster** should be **deployed** to **Google Cloud Platform**.

Example Workflow:

When you run your application on Google Cloud Platform, you are expected to get an interface that looks like the following:

```
Welcome to Big Data Processing Application
Please type the number that corresponds to which application you would like to run:
1. Apache Hadoop
2. Apache Spark
3. Jupyter Notebook
4. SonarQube and SonarScanner

Type the number here >
```

And when the user types a number (e.g. 3) in the terminal, the corresponding application (e.g. Jupyter Notebook) will run on the cluster.

Important Guidelines:

- You may reuse any docker images/containers built by others.
- Make sure that your entire application gets installed and prepared by the time you demand the user to enter one of the four options.
- No installations outside of Dockerfile (or Kubernetes cluster script) are expected to happen.
- Your application should be runnable without any custom/manual steps outside of launching the Kubernetes cluster along with Docker.
- You SHOULD NOT have any environment variables or configurations that are sat outside of your Dockerfile (or Kubernetes scripts) but you can use your Dockerfile and Kubernetes cluster scripts to set those variables.
- For Apache Hadoop, make sure to create one master node and two worker nodes.
- Use ReadMe.md file on your repository to list any assumptions, steps and any important information to share.

- Keep any private keys for your GCP account outside of the code on GitHub. Instead, submit them on Canvas along with your repository URL.

Submission Guidelines:

- Post URL for your GitHub repository to Canvas. Make sure to keep your GitHub repository public.
- You should complete this project individually. No group-work is offered for this project. However, you are welcome to share ideas. If you are using external references, refer to them.
- Your GitHub repository should have a ReadMe.md file that lists the "exact" steps on how to get this application to work. I will follow the steps in your ReadMe file and if I can't get it running on my machine, I will deduct considerable number of points from your project grade.
- You should record a video demonstrating two elements:
    1. Code Walkthrough while you are explaining your code changes.
    2. Demoing the running application while you are navigating through EVERY functionality that is working in your application. I will use this video to help assessing your grade. You may lose points for the functionalities that are not demonstrated in the demo.
- Your video size may be large to be uploaded to GitHub. You may use Box to upload the video and add the URL to your ReadMe.md file in your GitHub repository.
    1. Make sure that your video is publicly shared. Private videos won't be visible to the instructor and TAs and therefore, your project grade will be impacted.

Grading Criteria:

- Getting all four applications "containerized": 50% of the total project grade.
- Deploying all containers to Kubernetes Cluster: 25% of the total project grade
- Deploying Kubernetes cluster to Google Cloud Platform: 25% of the total project grade

- Extra-credit: building Graphical User Interface for this application: +20% of the total project grade.

Suggested Project Task Schedule (You may run ahead of schedule):

Week	Task
End of Week-4	Build the main terminal application.
End of Week-5	Create all docker images and containers.
End of Week-6	Study about Kubernetes clusters in Detail.
End of Week-8	Create Kubernetes Cluster and Configure it for the docker containers you Created
End of Week-11	1. Deploy your Kubernetes Cluster to Google Cloud Platform and work out any issues. 2. Record a video for the running version of the application with Code Walkthrough.
End of Week-12	1. Try to build the GUI for the Application. Otherwise, leave it with the terminal interface. If you got it to work, re-update your video recording with the code walkthrough. 2. Finish your ReadMe.md file to list all your steps, assumptions, and any information you find important to share.

Common Penalties:

- Your GitHub repository is not public: 100% reduction (won't be graded)
- Late submissions on Canvas or GitHub: 100% reduction (won't be graded)
- Not submitting the GitHub video (for both code walkthrough and functionality demo) or having the video not publicly shared: 20% penalty (calculated from maximum project grade).
- Not providing clear details in the ReadMe file on how to run the application (or any variables that need to be updated/replaced): 10% penalty (calculated from maximum project grade)