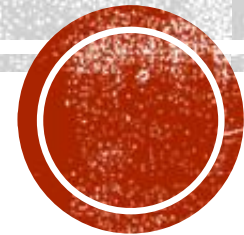
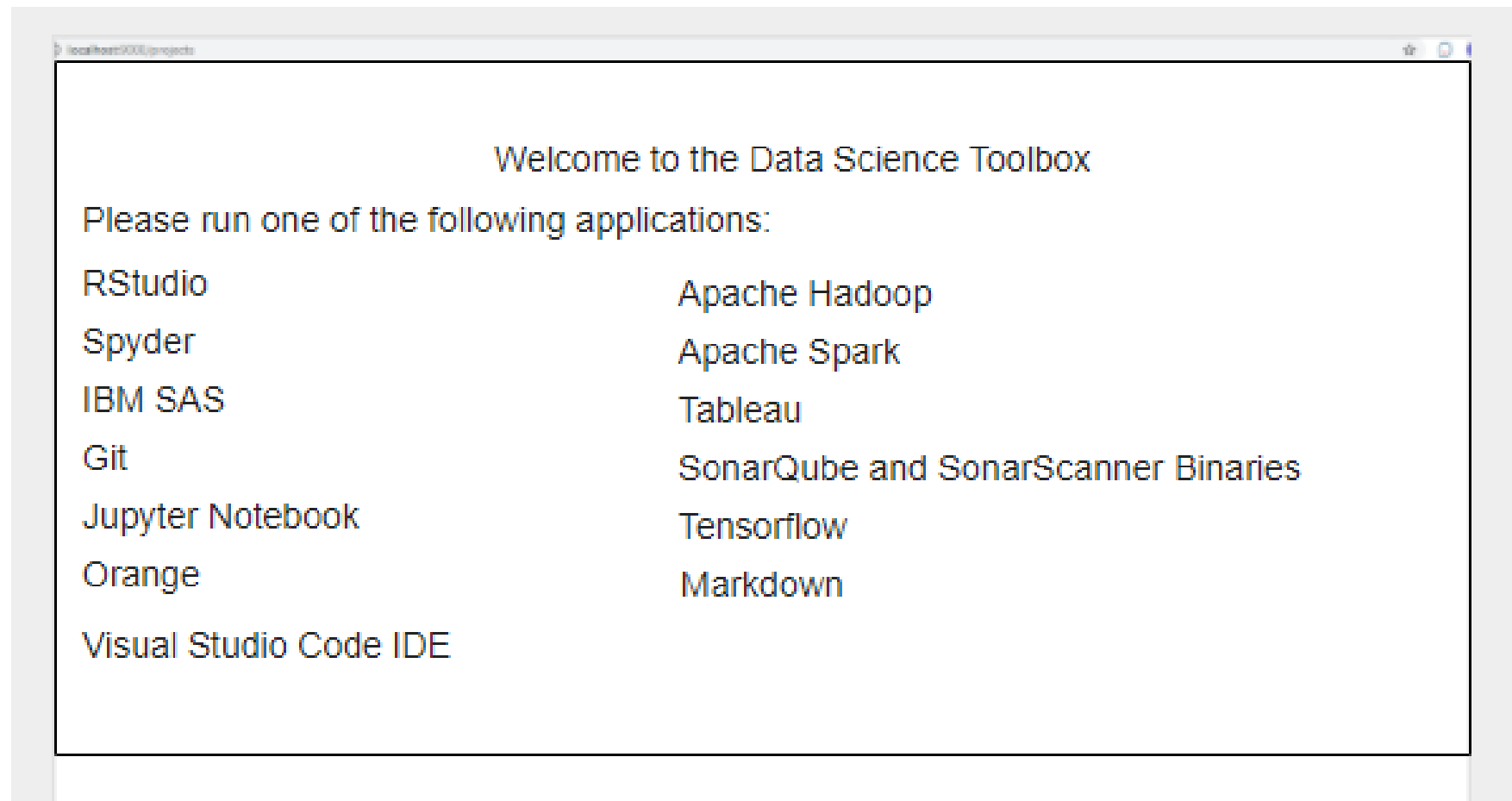


PROJECT-1 DESCRIPTION

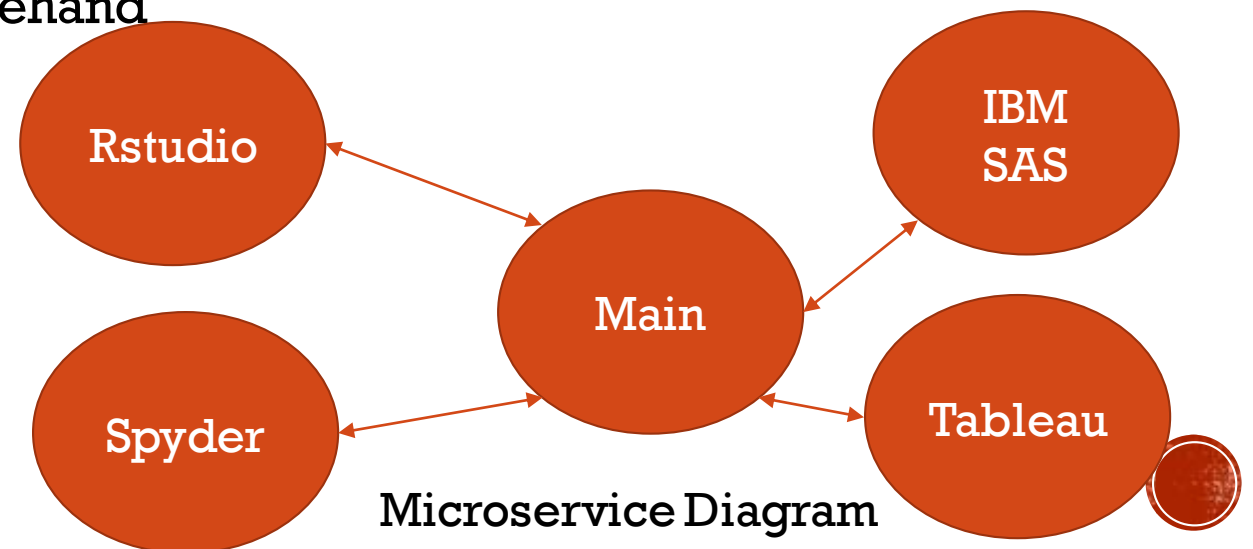


DATA SCIENCE TOOLBOX



PROJECT DESCRIPTION

- Build Microservices-based application where there is one main microservice that acts as the main application showing the User Interface in the previous slide.
- A demo video is provided on the course page. This demo shows one way of implementation and it doesn't aim to dictate how you should implement the project.
- Each option that is displayed represent a microservice application that is hosted on Docker.
- The user will click on one of the options and the selected application will be run.
- All applications should be **INSTALLED** beforehand



PROJECT GRADING DISTRIBUTION

- Main Application deployed on Docker: 15%
- RStudio: 5%
- Spyder: 5%
- IBM SAS: 5%
- Git: 5%
- Jupyter Notebook: 5%
- Orange: 5%
- Visual Studio Code IDE: 5%
- Apache Hadoop: 10%
- Apache Spark: 10%
- Tableau: 10%
- SonarQube & SonarScanner (or SonarCloud): 10%
- Tensorflow: 5%
- Markdown: 5%

Important Notes:

- Each application should be its own microservice.
- Each microservice should be deployed on docker container.
- Make sure that your microservice application gets installed through Docker – if needed-.
- No installations outside of Dockerfile are expected to happen.
- I should be able to run the entire application from your Dockerfile without any custom steps outside of Docker.
- If the user selects an application that requires installation, student won't get the grade for this application (i.e. your Dockerfile should install the applications before the application runs).
- You **SHOULD NOT** have any environment variable or configuration that is implemented outside of your Dockerfile but you can use your Dockerfile to set those variables



EXTRA CREDIT OPTION

Option	Extra Points
Make Spyder, Jupyter and Hadoop read/write files from your host filesystem (outside of your docker image)	2 points

- Notes:
 - You need to apply this extra-credit for all 3 applications. If you applied it for only subset of them, no points are granted.
 - If you implement this extra-credit, please help me by noting that you completed it in your ReadMe file. Unobvious/unmentioned notes are at-risk of not being caught by me.



PROJECT SUBMISSION GUIDELINES

- You should complete this project individually. No group-work is offered for this project.
- You should submit URL for your GitHub Repository containing the project code by April 7th, 11:59 PM EST.
- Your GitHub repository should be public. Private repositories won't be graded.
- Your GitHub repository should have a ReadMe.md file that lists the “exact” steps on how to get this application working on a new machine. I will follow the steps 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 application that is working in your application. I will use this video to help assessing your grade. You may lose points for the applications that are not demonstrated in the demo.
- Your video size may be large to be uploaded to GitHub. You may use OneDrive to upload the video and add the URL to your ReadMe.md file in your GitHub repository.
 - Make sure that your video is publicly shared. Private videos won't be visible by the instructor and therefore, your project grade will be impacted



PROJECT POSSIBLE PENALTIES

- This list contains a group of common penalties that may be applied:
 - Late submissions on Canvas or GitHub: **100% reduction (won't be graded)**.
 - GitHub repository is Private and course instructor doesn't have access to it: **100% reduction (won't be graded)**.
 - Not submitting the GitHub video (for **both** code walkthrough and functionality demo): **50% penalty**.
 - Not providing clear details in the ReadMe file on how to run the applications (or any authentication variables that need to be updated/replaced): **30% penalty (5 points)**



USEFUL LINKS

- Docker Compose: <https://docs.docker.com/compose/gettingstarted/>
- Filmora Wondershare Screen Recorder: <https://filmora.wondershare.com/>
- How to record video using Filmora: <https://www.youtube.com/watch?v=yjlyvvCOaMc>
- IBM SAS: <https://github.com/sassoftware/sas-container-recipes>

