Scenario Introduction:

In the dynamic landscape of modern web development and infrastructure management, automation plays a pivotal role in ensuring efficiency, reliability, and scalability. SwollenHippo Enterprises is looking to streamline its server environment setup, web application deployment, and ongoing maintenance processes across three distinct environments: Development (Dev), Testing (Test), and Production (Prod). As a DevOps engineer at SwollenHippo Enterprises, you have been tasked with creating an automated solution that not only configures the server environment but also handles the deployment of a web application from a Git repository in each environment.

Project Title: Automated Server Configuration, Deployment, and Maintenance for Multiple Environments

Scenario Details:

SwollenHippo Enterprises operates a two-tier architecture with a web server responsible for serving a simple static webpage and a database server managing essential data. Additionally, the organization is adopting a new web application that is hosted on a Git repository. The goal is to have a seamless and secure process that not only sets up and maintains the server environment but also ensures that the latest version of the web application is deployed automatically across the Dev, Test, and Prod environments.

Requirements:

- **Server Environment Setup:**
    - VM1: Web Server (Install Apache and Node.js)
    - VM2: Web Server (Install Apache and Node.js)
    - VM2: Database Server (Install MariaDB)
- **Ansible Playbooks:**
    - Create separate Ansible playbooks for each environment (Dev, Test, Prod).
    - Install necessary packages for both servers in each environment, including Apache and Node.js on the Web Servers and MariaDB on the Database Server.
    - Extend the playbooks to automate the deployment of a web application from a Git repository for each environment by calling a shell script (HINT: /var/www/html)
        - Git repo is available here: git@github.com:ttu-bburchfield/swollenhippofinal.git
- **Bash Shell Scripting:**

    - Write a bash script that automates the process of setting up the server environment and deploying the web application using the appropriate Ansible playbooks based on the specified environment.
    - The script should prompt the user for necessary configuration parameters specific to each environment. (HINT: branches matter on the Git repo)
    - Implement error handling and logging in the bash script.
- **Cron Jobs:**
    - Schedule a cron job to run the Ansible playbooks at regular intervals for each environment. (This is a new feature, but easy to do. Use a shell script that runs every minute)
    - Schedule a cron job to run an Ansible playbook that ensures the packages on the server (Apache, NodeJS, and MariaDB depending on server is the latest version)
- **Documentation:**
    - Provide clear and concise documentation on how to run the bash script, configure cron jobs, and update the server environment using Ansible playbooks for each environment.
    - Include documentation on how to configure the Git repository for the web application deployment for each environment.
    - Include troubleshooting steps and common issues.
    - Document your scripts and Ansible playbooks

Evaluation Criteria:

Your project will be evaluated based on the criteria mentioned above. This is a ranked graded assignment meaning those that perform the best will get the highest score. This does not mean that everyone gets an A nor does it mean not everyone will get an A.

Submission:

Submit a zip file containing all relevant scripts, playbooks, documentation, and any additional files needed to run the project, including documentation on Git repository configuration for the web application for each environment. When developing your deliverables keep in mind that it should include EVERYTHING that another user will need to get your process to work.