LETTERKENNY INSTITUTE OF TECHNOLOGY

ASSIGNMENT COVER SHEET

To Be Completed By The Student

Lecturer’s Name: Ruth Lennon

Assessment Title: Broken Legacy code

Submission Date: 25-11-2022

Student’s Name: Darshan Pawar Id. Number:l00171222\_

Course / Stage

Subject/Module: Iac for DevOps

Word Limit: Actual Word Count: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I confirm that the work submitted has been produced solely through my own efforts.

Student’s signature: Darshan Pawar Date: 25-11-2022

|  |
| --- |
| **Note**: **PENALTIES**   * The total marks available for an assessment will be reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. * Assessment work received more than two weeks late, without prior approval by the lecturer will receive a mark of zero. * Marks awarded will be reduced by 10 % if submitted work is greater than 10% above or below the assigned word limit. * A further hard or electronic copy of your submitted work may be requested, and therefore you must keep a copy on disc. * Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute’s Assessment Regulations   **Plagiarism:** Presenting the ideas, words of someone else without proper acknowledgement. Refer to the Institutes’ procedures and guidelines for the assessment of learners. |

DevOps Implementation:

We must first adopt the CALMS framework to implement the DevOps Principles and methodology in the organization. CALMS stands for Culture, Automation, Lean, Measurement, and Sharing.

Culture - Change of Culture referred to as ‘Left-Shift’ – consider quality earlier

Automation - Implementation of Automation to speed up the process, thereby ensuring more frequent software releases

Measurement - Using Measurements will help determine if progress is being made in the intended direction. Metrics are important for comparing

Lean - eliminating low-value activities.

Sharing - Frequent/Continuous release means that we have higher visibility of the product we share, along with enabling the customer to indicate where changes are needed early in the cycle.

Initially, we must make strong bonds between development and operators’ teams with DevOps culture and mindset. Our team must cohesively work together and understand the needs and expectations of all members. In a small team, this is easy to implement along with everyone gets an opportunity to make an update.

Automating the Development, testing, server configuration, and deployment is most important in the organization. Because of automation, we don’t need to do repetitive tasks. Using CI/CD pipelines we can develop and deliver product faster.

Tracking the development of DevOps flow operations is crucial for achieving consistent and ideal performance. The system's key performance indicators may be measured to reveal what functions well and where improvements can be made.

Since the time bound is the important factor in our case, we must eliminate low-value tasks and irrelevant steps. In DevOps, the concepts of continuous improvement are considered opportunities. In an organization, collaboration is an important aspect.

Sharing feedback, knowledge, and best practices promotes clarity and eliminate constraints.

Process for modernizing:

To make the existing code more efficient and maintainable, the Code refactoring process is used. The QA and debugging processes run considerably more smoothly as a result of code refactoring, which increases readability. Furthermore, while it doesn't get rid of bugs, it can undoubtedly aid in avoiding them in the future. The best code refactor approach is Red-Green-Refactor.

Chart, diagram, schematic

Description automatically generated

Red:

The Red phase is starting phase of the Red-Green-Refactor cycle. The red phase indicated which code steps were causing errors, and we had to stop and check what causing the failing test.

Green:

In the Green phase, we must solve the failing test in the red phase. Here, we must focus on solving the problem rather than on optimization or maintaining the code structure. After the test is passed, we can focus on optimizing the code.

Refactor:

In Refactor phase, we need to think about implementing the code better or more efficiently.

Another thing we can use Cloud-Native Technology to host applications in the Cloud platform to avoid downtime, and if the issue arises in the system, it can be fixed anywhere. The technologies in cloud platforms advance quickly, providing development and operations teams with fresh approaches that foster innovation. We can also use CI/CD pipeline to improve product delivery. Also, this method will make development and infrastructure teams to work in partnership. Improvements in CI/CD are important to provide fast feedback from customers, find security vulnerabilities and resolve code issues.

In the legacy system application and technologies, some of the things are redundant and irrelevant in modern architecture. We can eliminate that previous application component with new requirements. Also, some of the methods can be automated in the legacy system using toolsets and new technologies. Furthermore, we can find errors using unit testing and refactor them using testing tools.

Review the code to demonstrate the modernization process:

Using code refactor methods like Red-Green-Refactor, we can find errors in codes that can be fixed using the code refractor method. It will also make code more maintainable and efficient. Code Refractor will increase the code readability, reliability, and reusability. Using cloud platforms and CI/CD integration tools, we can increase team collaboration and solve the code problem quickly with the continuous integration of customers, resolve security vulnerabilities and resolve code issues.

To remove redundancy and irrelevancy in code, instead of updating the IPAddresses.txt file with new IP addresses or Computer Names, we can use IP configuration command (ipconfig) script command, which can be executed before opening the script, to automatically inherit the IP address instead of changing the file every time. We can make a script that will automatically enable PS Remoting and Windows Remote Management, configure WinRm, and Run the Test-WsMan test instead of configuring the server. List of $computerNames we can find in the logs file and can be sent into an array for loop. A Dynamic reference or variable can be added instead of computerName. We can use a foreach loop to go through all $ports in $PortList.

Conclusion:

As DevOps Engineer, my responsibility is to implement the DevOps methodology in the organization. The current downturn in the economy left the company in dire straits. Because of this, organizations to take modernize and streamline ideas in their products and services. With the tight time bounds, I have proposed DevOps methodology in this assignment to justify working on my decision.

Since the CALMS framework is easy to assess an organization to adopt the DevOps process, and the organization has already been using the long legacy methodology I thought CALMS would be easy to implement to gradually start in the organization. CALMS frameworks consist of Culture, Automation, Lean, Measurement, and Sharing.

With the time-bound constraint, this framework will be easily manageable as it’s mostly focused on collaboration, like team joining daily sprint stand-ups and planning sessions to work out each other’s ideas, work, and struggles. Automation involves the elimination of repetitive work and creates a resilient system. Automation I can implement in an organization using DevOps tools and introduce CI/ CD pipelines. We will remove redundant and irrelevant processes using the lean methodology and be more agile. Measurement will make me understand the organization's current logs and task flow so I can make changes and take decisions accordingly.

In my Team, I will use Ren to understand the Legacy code structure and implementation, which will also help me train Jalen to refactor code using the new modernization method. I will also use Cloud Infrastructure and Cloud-Native Technology to avoid unnecessary downtime. To remove redundancy in code I have proposed some code reviews.

In the time-bound constraint, legacy code can be refactored using the Red-Green-Refactor method, which will be easy to implement and low risk of application failure.

With the DevOps principles, new modern strategies will help the organization in the development and deployment phases and the business plan.