|  |
| --- |
| **BAHRIA UNIVERSITY (KARACHI CAMPUS)**  **ASSIGNMENT # 4 – SPRING 2022**  **Cloud Computing (SEN-325)**  **[CLO 4]**      Class: **BSE-6 (A/B)**  Max Marks: **3**    Course Instructor: **Engr. Muhammad Faisal** Submission Date: **18th June, 2023** |

**RESEARCH REPORT ASSIGNMENT**

In this assignment, the students will prepare a research report of the topic selected in the assignment 3. The group will be same as presentation group.

Report should contain following headings.

* **Abstract.** [Should contain the abstract/summary of your report]
* **Introduction**. [Will introduce all the technologies related to selected topic]
* **Review of Literature**. [Will discuss the existing work done in selected topic]
* **Conclusions** [Final wording]
* **References** [References of the research papers and websites used to develop the report]

|  |  |
| --- | --- |
| **Sr#** | **Title of the paper** |
| 1 | Green Cloud Computing |
| 2 | Edge Computing |
| 3 | Cloud Cryptography |
| 4 | Microsoft Q# and Azure Quantum |
| 5 | Cloud Analytics |
| 6 | IoT hub |
| 7 | DevOps |
| 8 | MLOps |
| 9 | Mobile Cloud Computing |
| 10 | Implementation of Big Data Analytics using Cloud Computing |
| 11 | AR Cloud (Augmented Reality) |
| 12 | Benefits and Applications of Blockchain in Cloud Computing |

To submit this assignment, the deadline is:

**18th June, 2023**

If you have any query, feel free to contact at: **mfaisal.bukc@bahria.edu.pk**

Page **1** of **1**

**Topic: “DevOps”**

**Summary:**

DevOps is a software development methodology that emphasizes collaboration, integration, and automation between development and operations teams. It consists of several phases or stages: Plan, Code, Build, Test, Deploy, Operate, Monitor, and Iterate. These phases are not strictly sequential but overlap and interact with each other throughout the software development lifecycle.

DevOps tools play a crucial role in supporting and streamlining different aspects of the DevOps process. They include version control systems like Git, CI/CD tools like Jenkins, configuration management tools like Ansible, containerization platforms like Docker, monitoring tools like Prometheus, and collaboration tools like Slack.

**Introduction:**

DevOps is a software development methodology that aims to bridge the gap between software development (Dev) and IT operations (Ops) teams. It promotes a collaborative and integrated approach to software delivery, focusing on improving communication, collaboration, and automation across the entire software development lifecycle.

Traditionally, development and operations teams have operated in separate silos with distinct goals and priorities. This often led to inefficiencies, delays, and misalignment between the two teams, resulting in slower time to market, reduced software quality, and increased operational challenges.

DevOps breaks down these silos and encourages close collaboration between developers, testers, system administrators, and other stakeholders involved in the software delivery process. By fostering a culture of shared responsibility, continuous improvement, and automation, DevOps aims to streamline and accelerate the development, deployment, and operation of software applications.

One of the fundamental principles of DevOps is to automate repetitive tasks and eliminate manual processes wherever possible. This includes automating code builds, testing, deployments, and infrastructure provisioning. By automating these processes, teams can achieve faster, more reliable, and consistent software delivery, reducing errors and increasing efficiency.

Another key aspect of DevOps is the adoption of a continuous integration and continuous delivery (CI/CD) approach. Continuous integration involves regularly integrating code changes into a shared repository, where automated tests are run to detect issues early on. Continuous delivery focuses on automating the deployment of tested code to production-like environments, ensuring that the software is always in a deployable state.

DevOps also emphasizes the importance of monitoring and feedback loops. By continuously monitoring the performance and health of applications and infrastructure, teams can proactively identify issues, address bottlenecks, and optimize resources. Feedback loops, involving stakeholders and end-users, help drive improvements, gather insights, and align software development with the evolving needs of the users.

The adoption of DevOps practices and principles has gained significant momentum in recent years, as organizations seek to improve their agility, speed of delivery, and overall software quality. It enables organizations to respond more effectively to market demands, reduce time to market, enhance collaboration, and deliver higher-value software to end-users.

**Review of Literature:**

We discussed the phases of DevOps, which include planning, coding, building, testing, deploying, operating, monitoring, and iterating. Each phase plays a crucial role in the software development lifecycle and contributes to the overall success of DevOps implementation.

We also covered various categories of DevOps tools, such as version control systems, continuous integration/continuous delivery (CI/CD) tools, configuration management and infrastructure as code tools, containerization and orchestration tools, monitoring tools, collaboration and communication tools, and security tools. These tools are instrumental in automating and streamlining different aspects of the DevOps process, enabling efficient software development and delivery.

Moreover, we highlighted the advantages of adopting DevOps practices. These advantages include faster time to market, increased collaboration and communication, improved software quality, enhanced reliability and stability, faster recovery from failures, scalability and flexibility, cost optimization, and continuous improvement. These benefits collectively contribute to increased efficiency, agility, customer satisfaction, and overall business success.

**Conclusion:**

In conclusion, DevOps is a transformative approach that brings together development and operations teams to foster collaboration, streamline processes, and automate tasks throughout the software development lifecycle. By breaking down silos and promoting a culture of shared responsibility and continuous improvement, organizations can achieve significant benefits.

The phases of DevOps, including planning, coding, building, testing, deploying, operating, monitoring, and iterating, provide a structured framework for teams to follow. DevOps tools in various categories facilitate automation and support different aspects of the DevOps process, empowering teams to work efficiently and deliver high-quality software.

The advantages of DevOps are wide-ranging and impactful. Organizations can experience faster time to market, increased collaboration and communication, improved software quality, enhanced reliability and stability, faster recovery from failures, scalability and flexibility, cost optimization, and a culture of continuous improvement. These advantages enable organizations to respond to market demands, deliver value to customers, and stay competitive in the fast-paced world of software development.

Overall, DevOps represents a paradigm shift in software development and operations, driving efficiency, agility, and customer satisfaction. By embracing the principles of collaboration, automation, and continuous improvement, organizations can unleash the full potential of their software development teams and achieve greater success in delivering high-quality software products.

**References:**

1. https://aws.amazon.com/devops/what-is-devops/
2. <https://www.atlassian.com/devops>
3. "The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford.
4. "DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations" by Gene Kim, Jez Humble, Patrick Debois, and John Willis.
5. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley.
6. "Site Reliability Engineering: How Google Runs Production Systems" edited by Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy.
7. "Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations" by Nicole Forsgren, Jez Humble, and Gene Kim.