Name :- Vaishnavi Nagnekar
Batch
~~ROLLNO~~ :- T13
ROLLNO :- 66

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

ASSIGNMENT - 02

SOFTWARE ENGINEERING & PROJECT MANAGEMENT

To understand DevOps :- Principles, Practices & DevOps Engineer Role & Responsibilities.

What is DevOps?

DevOps is a collaborative approach where team work together to build & deliver secure software efficiently. It combines software development (dev) & operations (ops) to accelerate delivery through automation, collaboration, fast feedback & Iterative improvement. Built on Agile methodology, DevOps creates a culture of accountability, collaboration & shared responsibility for business outcomes

core Principles of DevOps :-

Develop & test in production - like environments
* Develop builds frequently
# continuously validate operational quality.

Key Practices of DevOps

1. continuous ~~Development~~ Deployment

continuous delivery & deployment originate from continuous integration, a method to rapidly develop, build & test new code with automation to that only code that is known to be good becomes part of a software product.

2. Continuous Development
This is the phase that involves planning & coding, versioning & managing builds of the software applications functionality. Eg:- Git, Github, Maven.

3. Continuous Testing :-
continuous testing is executing automated tests, continuously & repeated against the code base & the various deployment environment. It is a software testing methodology which focuses on achieving continuous quality & improvement.
Eg :- Appium, Bamboo.

4. continuous Integration :-
continuous Integration refers to the build & unit testing stages of the software release process. Every revision that is committed triggers an automated build & test
Eg :- Tenkins, Travis, CI

5. Infrastructure Management
Without automation, building & maintaining large-scale modern without automation. IT systems can be a resource intensive undertaking & can lead to increased risk due to manual error. Configuration & resource management is an automated method for maintaining computer systems & software in a known, consistent state.

6. Configuration Management.
Infrastructure as code is the practice of describing all software runtime environment & networking settings & parameter

in simple textual format that can be stored in your version control systems (VCS) & versioned on request. These text files are called manifests & are used by DevOps tools to automatically provision & configure build servers, testing, staging & production environment.
Eg:- chef, salt-stack.

DevOps Engineer Role:-
A DevOps engineer manages a company i.e IT infrastructures, bridging development & operations the primary goal is to improve the process & efficiency throughout the software development lifecycle.

Key Role:-

1. Facilitators of collaboration:-
Bridging the gap between development operations & QA teams to streamline communication.
2. Automation Specialist
Automate repetitive tasks like testing deployment & monitoring.
3. continuous Integration & Continuous Delivery (CI/CD):-
Design, implement & maintain CI/CD pipelines to enable faster, reliable & repeatable software releases.
4. Infrastructure as code:-
Use tools like Terraform, Ansible or cloud formation to define & provision infrastructure through code.
5. Monitoring & Incident Management.
Set up monitoring system to track application performance & troubleshoot issue in real time. It also ensures that systems are resilient & downtime is minimized.

6. Cloud & Infrastructure Management
Deploy, manage & optimize applications on cloud platform like AWS, Azure or Google cloud also handles container orchestration

Key Responsibilities :-

1. Collaboration & Planning :-
Work with development & operations teams to plan & design scalable solutions
2. Configuration Management.
Uses tools like Puppet, chef or Ansible to manage server configuration & ensure consistency
3. Pipeline Management.
Maintain CI/CD pipelines to ensure seamless build, test & deployment workflows.
4. Monitoring & Logging :-
Implement monitoring tools like prometheus, curjana or splunk to track system health & measurement performance.
5. Support & Troubleshooting.-
Respond to incidents & resolve production issues promptly & identify root causes of failure & implement fixes.
6. Documentation & Reporting
Document system configurations, deployment processes & troubleshooting guides.