DENNIS KIMUTAI KIMAIYO

S223224152

DEVOPS CYCLE

1. Plan

This stage entails the definition of project goals, requirements using use case, allocation of resources e.g. time, skilled labour and time, creation of a roadmap and scheduling of development.

Tools - Jira or Trello for project management and Microsoft teams for communication.

Automation of task scheduling and backlog prioritization using tools like Jira.

Feedback/Monitoring - Capture requirements changes, track task progress, and analyze project timelines.

2. Create

This is the immediate stage after the planning stage. This stage includes developing and writing code based on the requirements defined in stage one. Code merging, quality and performance are assessed at this stage to ensure they meet the standards and are ready for integration.

Tools - Git, GitHub.

Code reviews, syntax checking, and static code analysis are automated at this stage.

Feedback/Monitoring - Code changes, and quality metrics

3. Verify

Various types of testing are done on the code to identify bugs and verify functionality. This is to ensure the code works as expected and meets quality standards. the tests include acceptance, regression, configuration, quality analysis, quality analysis, configuration, performance and release test.

Tools - Jenkins, Travis CI, CircleCI, Selenium.

Automate continuous integration, and build processes.

Feedback/Monitoring - Test results, build status, code coverage reports, and bug tracking.

4. Preprod

This stage is all about preparations for the application release. The final tests and

validations are conducted to ensure readiness for release.

Tools - Docker, Kubernetes, Ansible.

Automation: Automated deployment scripts, containerization, and orchestration using Docker and Kubernetes.

Feedback/Monitoring - Deployment logs, performance indicators from the staging environment, and the outcomes of the final validation.

5. Release

Deployment of the application to the production environment. This ensures a smooth transition from development to live usage.

Tools: Jenkins, GitLab CI/CD, Spinnaker.

Continuous deployment pipelines, rollback mechanisms, and release management are automated.

Feedback/Monitoring - Release notes, deployment logs, production readiness checks.

6. Configure

Define and apply infrastructure provisioning and configuration, including storage, database, and network. Ensure consistency and security.

Tools - Configuration management databases (CMDB).

Feedback/Monitoring - configuration changes, environment consistency checks, and infrastructure health metrics.

7. Monitor

In this phase, the application is monitored and maintained in the production environment. This includes tasks such as monitoring performance, tracking user behavior, and troubleshooting issues.

Tools - ELK Stack (Elasticsearch, Logstash, Kibana), Nagios.

Automate alerting, log aggregation, and anomaly detection.

Feedback/Monitoring - Real-time performance metrics, system logs, incident reports, and user feedback.

The DevOps cycle integrates development and operations to enhance collaboration and efficiency. These stages collectively ensure a structured approach to software development, from planning and creation to deployment and monitoring, aiming for

quality, efficiency, and continuous improvement. Monitoring and feedback mechanisms are integral to each stage, enabling continuous improvement and quick response to issues.

REFERENCE

1. https://testsigma.com/blog/devops-life-cycle/
2. Aiello, B., & Sachs, L. (2016). Agile application lifecycle management: Using DevOps to drive process improvement. Addison-Wesley Professional.
3. Armstrong, S. (2016). DevOps for networking: Boost your organization's growth by incorporating networking in the DevOps culture. Birmingham, UK: Packt Publishing.