Birla Institute of Technology and Science, Pilani

Introduction To DevOps

DevOps Critique Report

Team No: 2

Sl. No	Team Members	BITS ID
1	Avinash Rao	2020HS70019
2	Utpal Gaurav	2020HS70052
3	Aditi Sahai	2020HS70009
4	Abhijeet Kumar	2020hs70044
5	Gourab Mukherjee	2020hs70041

Introduction

Project Name: - Donation Drive

Purpose: - A platform for raising funds for a medical emergency or social cause by gathering voluntary financial contributions by engaging individuals, businesses, charitable foundations or governmental agencies.

Summary

The objective of this document is to critically analyze Devops practices followed during the development of the given project. We will analyze various aspects of Devops like: -

People Aspect: - How different roles and responsibilities were assigned to people to achieve maximum productivity.

Process Aspect: - What processes we followed to have a clear goal and objective to achieve continuous collaboration.

Product Aspect: - Tools, Software and Technologies used to achieve highest efficiency.

Critical Evaluation

People Aspect

- Technical expertise of each member was identified, and tasks were allotted accordingly.
- Each team member was given autonomy to complete their respective tasks in their own way.
- Roles like scrum master, developer, tester were assigned to appropriate people.

Process Aspect

- The problem statement was analyzed and broken into relevant bugs, stories, epics and other tasks in sprint planning meetings and Jira items were created accordingly.
- Scrums were held every alternate day to check the progress.
- Weekly sprint review was done where each team member demonstrated the work done to the team.
- Sprint retrospective was not done which should have been followed for better transparency in the team.

- Peer review was practiced improving code quality of project with mutual understanding of the code.

Product aspect:

- GitHub was used for version control to have a single source of truth.
- Continuous Integration -
    - Each pushed commit will be checked for proper lint before passing.
    - Unit test setup was done to ensure reliability by Jenkins.
    - Sonar was used to check code quality, maintainability and coverage.
    - Commits were built individually by Jenkins before integration with the main trunk.
    - GitHub Jenkins webhook was not implemented due to unavailability of Jenkins instance with external Ip address.
- Continuous Delivery
    - Each microservice was dockerized so no platform dependency could seep-in.
    - Every run of main trunk builds docker images of each microservice and pushes to respective docker repositories in dockerhub.
    - All secrets are handled properly with Jenkins, and none are hardcoded.
    - Versioning of docker images wasn't implemented which is an essential feature to decide release candidate in production.
    - Acceptance testing was setup but not automated in the pipeline (future enhancement).
- Continuous Deployment
    - Gardener tool hosted by SAP was used to provision a Kubernetes cluster.
    - Kubernetes charts were created so it gives the following benefits out of the box: -
        - Microservices are highly scalable.
        - Pods are self-healing.
        - Container to container communication.
        - Load Balancing is handled internally by ingress and other services.
- Continuous monitoring
    - ELK is not implemented in the tech stack for now. (Future enhancement)
    - Observability tools like Dynatrace could have been implemented for better monitoring. It would provide application to infrastructure level visibility.

Challenges faced

There were many challenges faced by our team during the creation of this project but after the proper implementation of agile methodologies in our team we were able to manage all the tasks and planning that were required for the proper completion of the project.

The First challenge faced was the architecture creation of our project and what tools will be best suited as per the requirements of our project. For this we organized regular planning calls where we brainstormed, discussed and selected the best tools required for our project.

The other challenge that we faced was the distribution of tasks. For that we distributed tasks to modules so that each member can contribute and work on modules independently and test it without any dependency on other's tasks.

Learning various new tools was difficult. Since most of us never implemented the devops tools there was a high learning curve that we need to go through. But we took this as a challenge and learned it for the implementation.

Enhancements to current practices

- Every team member should always write unit tests for the feature they are implementing before pushing the code to the central repository (Test Driven Development).
- Every team member should be proactive in testing features and raising bug items in Jira.
- In each pull request other developers should give proper reviews and they should be fixed by the developer.
- PR validation is not implemented, this should be there so that we will be able to make sure that the new feature is not breaking anything.
- The branches should not be long lived and should be integrated.
- Each developer should first pull the code before starting on a new feature.
- Failing Jenkins build of main trunk should be fixed with top priority.
- Alerting can be introduced through emails or other channels in case of build failures.