Devops Critique of our Project

Submitted by -

Kanav Kamat – 2020HS70008
Pooja Hegde – 2020HS70026
Sohan Kulkarni – 2020HS70051
Rakesh Aithal – 2020HS70032
Zaid Tergaon – 2020HS70031

Project Description

This project mainly focuses on new product development and feature enhancement. The developers develop the features and push it to the git where there are different checks and rules to be checked on the new code, once the code is properly tested and checked the merge of code is allowed.

These rules and checks on the code are set by Devops team to standardize the code format and reduce the errors and issues in production.

Team and Tools

The team takes care of all the environments, pipelines, deployments, and releases. This team majorly investigates the pipelines and release related issues that the development team is facing.

There are some standard tools used by team for monitoring, integrating, and deploying. These are the tools

1. **Git and Github** – This is a major tool used by development team and Devops team. This tool is majorly used for collaboration on the code and maintain version control. Mainly Devops team lays some rules and checks in github before a developer is allowed to merge a code. This is a way how Devops team try to standardize a code quality and reduce the risk of falling during release and in production.

2. **Sonar cloud** – This is tools used by the team to check if the code has any vulnerabilities or bugs in it. This is a tool used to check the code quality of the repository. This can be helpful to find the unwanted vulnerabilities and test coverage. There is a practice in our team that test coverage of our code should be always higher that 80%, and this tools really helps us in knowing the test coverage.

3. **Jenkins** – this tool is used for continues integration. Our team has created pipelines for various checks. These pipeline check for latest builds, check for the code lint, check if tests are done properly and other criteria. This is also used to manage the global configuration of the project which includes plugins pipeline scripting helper etc.

4. **ELK stack** – This team uses this stack to do the monitoring of errors bugs and logs. ELK stands for elastic search, Logstash and Kibana. This stack

has been integrated with our github repository. So, when there are any issues in code and found and security logs these are shown in our Kibana dashboard. Kibana dashboard helps in visualizing and utilize various aggregate functions to present the logs and errors in much more meaningful way.

5. **Docker** – This tool is basically used to containerize the application with all its dependencies and ship it to customer. This is also used to setup a basic local machine for any new developer and help him with the development process. This tools majorly helps in packaging the entire application and shipping it to the customer to run seamlessly.

6. **Kubernetes** – This tool is used to maintain the various containers created by the docker and scale, monitor and orchestrate the application. This helps in load balancing and networking between the containers.

Challenges and Learnings

Challenges

- ELK setup took a lot of time. We faced a lot of issues while setting up the ELK stack in our AWS instance. The installation and getting the exact logs for our project was bit of tedious task. We faced issue in getting the logs into kibana dashboard.

- Setting up the Jenkins Pipeline. We had a lot of brainstorming session regarding the jenkins pipeline. We thought to at most of the plugins that would help us to standardize the code and make a successful cut for a release. We thought of different senecios to set up the pipeline with all the checks and rules.

- Understanding the deployment process. We had never worked on docker and Kubernetes. So, it was a challenge for us to make proper docker containers for the application where the code is evenly and independently distributed. Also using Kubernetes was a big task as it must be used to scale our applications and containers

Learnings

Basically, we have learnt a lot from our challenges that we faced

- ELK: We had no experience in this stack. Enabling monitoring for our application was a task for us and we faced many challenges as well while integrating it. Once we learnt the process of installation and integration to our project it was easy for us to monitor our logs and visualize the data.

- Pipeline: Creating a fresh pipeline for interesting task for us, as it included a lot of brainstorming and collaboration. We wanted to make a pipeline which has all the checks and rules to be implemented to have a proper code standard. This took us a bit of time to build and integrate it, but this was the most interesting task for us

- Deployment: no one from team had a hands-on experience on deploying an app and maintaining. So, each of us had to go through a lot of documentation and trial and error process to deploy an app. Once everyone was clear with process, we tried deploying our app and scaling which took lot of time and faced multiple errors well, but at the end it made us clear with the concept of deployment

Future Enhancements

Environments

Our team majorly focuses on use of multiple environments for various testing and staging. Before releasing it to the production the code must go through multiple staging environments for various kind of testing. Some of the environments are

- **Dev Environment** – used by developers to test and check the functionality before committing the code

- **Test Environment** – used by testers to test the new features and find the bugs in present

- **preProd Environment** – used to check all the features and developments before the release.

- **Production Environment** – the final stage of release where the customers can access the new features

Devops practices:

This Team Practices agile methodology. We have daily scrum/stand-up, weekly sprint meeting and monthly retrospect of the takt. Additional to these all fixes and changes have JUnits/QUnits which are suggested and presented during our daily Topic Syncs where team has a whole sit and solves blockers and long waiting problems.

CI/CD [continuous integration and deployment]

All the manual works are converted into automation now, instead of using Jenkins we are planning to use github actions as well which is better in performance and time of response. We use docker to containerization the applications and deploy it to the cloud at the same time we use Kubernetes to maintain all the container created by docker and scale them.