Intelligent Systems and Computer Technology D.J. Hemanth et al. (Eds.) © 2020 The authors and IOS Press. This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0). doi:10.3233/APC200198

Efficient Data Migration Method in Distributed Systems Environment

NatheeswariN^{a, 1}, SivaranjaniP^b, Vijay K^b and VijayakumarR^b

^aDept of Computer Application, PSG college of Technology, Coimbatore, India ^b Dept of CSE, Rajalakshmi Engineering College, Chennai, India

Abstract: Database Migration is a segment of a business application arrangement process wont to refresh changes in customer databases. This is frequently a significant procedure in application condition. Subsequently every association applies inventive information relocation strategies for refreshing their customer's databases. These updates are typically more steady in nature than the intermittent updates performed under conventional discharge rehearses. Visit little updates make every movement less unsafe. Be that as it may, just in the event of tremendous updates, relocation process hinders prompting off base and inadequate changes. During this task, the HRAPP application framework's database refreshes are moved utilizing another dependable and tedious strategy. We applied Continuous Integration and Continuous Deployment strategy for information relocation. Nonstop joining and arrangement pipeline technique might be a product improvement practice where designers consistently blend their code changes into a focal vault, after which computerized fabricates and tests are run. Here, arrangement code changes are consequently assembled, tried, and prepared for a discharge to creation. These movement documents are taken care of consequently through the sending pipeline, and changes to the databases are refreshed bolstered the relocation record. Right now, talk about how effectively database relocation are frequently executed with ceaseless joining and organization pipeline technique.

Keywords. HRAPP Continuous Integration, Continuous Deployment, Database migration

1. Introduction

Database relocation happens when there's a necessity to change database seller, overhaul database software, content of database tables or move an information to the cloud [2]. While moving information to the cloud, update conveyed databases, or return database substance, relocation is performed automatically by utilizing a movement instruments like Phinx, Doctrine, Flyway and so forth during this procedure, application frameworks need to deal with information without changing the database patterns. Some key errands incorporate surveying the database size to work out what extent stockpiling is required , testing applications and ensuring information using continuous integration and deployment pipeline method. Proposed solution is straightforward to implement for database migrations in PHP application systems.

¹Natheeswari, Deptof Computer Application, PSG College of Technology, Coimbatore, India E-mail: nnatheeswari@gmail.com.

where migrations are handled during deployment. Phinx tool supports migrations written in both PHP and SQL. Using PHP syntax, it's easy to implement migrations portable between different database vendors or hand-code advanced queries.

2. Existing System

At present, application system databases updates and modifications are handled using shell scripts. In application systems, number of clients varies and every client will have a database. Using script files, the database manipulation is performed manually by executing the script file within a loop by calling the distributed client databases.



Figure 1. Fully independent database pipelines

Fig. 1 illustrates fully independent database pipeline process. Currently database automation or integration is completed by building a separate pipeline dedicated to individual database, so as to avoid impacting existing application pipeline. Database and application have completely independent deployment pipelines, different code repositories, different Continuous Integration process and different Continuous Deployment process [6].

3. The Project

3.1 HRAPP

HRAPP may be a 'Software as a Service' (SaaS) product developed by Caprice Technologies Pvt Ltd, Coimbatore. HRAPP is a Cloud-based Human Resources Management System (HRMS) and offers comprehensive functionalities within the field of HR Management and Payroll Management. HRAPP is straightforward and straightforward to use dashboard-type interface which will be mastered in on time by individuals having basic computer knowledge and application skills. HRAPP empowers the HR team to hold out complex HR Management tasks like accessing details associated with organization, Employee's Data, Payroll, HR reports etc. HRAPP helps HR Team of the Organization to possess complete control over all the HR activities The database for the merchandise (HRAPP) is handled on the cloud, and changes in database are manipulated using migration script file, which is out there in system root folder. Initial deployment are going to be wiped out application environment and once testing is complete, final deployment are going to be implemented in production environment.

3.2 PHINX

Phinx allows developers to change and manipulate databases during a convenient way. It avoids SQL scripts and instead offers a strong API for creating migrations using PHP code. Phinx is installed using Composer, which may be a tool for dependency management in PHP applications.

3.3 DevOPS

DevOps (Development and Operations) innovation advances coordinated effort among Development and Operations Team to send code underway condition quicker in a programmed and repeatable manner [3]. It helps in expanding quicker application conveyance and administrations. In basic words, DevOps are frequently characterized as an arrangement of advancement and IT activities with better correspondence and joint effort It is a software development approach which involves continuous development, continuous testing, continuous integration, continuous deployment and continuous monitoring of the software throughout its development life cycle. The primary practice of DevOps is Infrastructure automation and Continuous Delivery. The tool used here for development and integration is Gitlab and it's elaborated below

3.4 GITLAB

GitLab is that the premier application management system for entire DevOps lifecycle. From getting to monitoring applications, GitLab covers every stage of the lifecycle and manages securely across stages. It helps teams manage and optimize their software delivery lifecycle with metrics and gain insight into how the business is performing. It enables portfolio planning and management through epics, groups, and milestones to arrange and track progress. It also helps teams design, develop and securely manage code and project data from one distributed version system . It enables rapid iteration and delivery of business values.

4. Proposed System

Data must be continued previously, during and after the sending. Rolling back a database is riskier than rolling back an application. The database changes are handled within the deployment pipeline in Gitlab The top result's that databases become a bottleneck in an agile delivery process.During this undertaking, the test suit considered for improvement is that the HRAPP database is moved from Mysql database to AWS distributed storage. Database and application code live in same store and offer same Continuous Integration and Continuous Deployment forms. Code for updates and changes in database are composed during a relocation document and condition setups, database name will be spared during a .yml record. From Mysql database to AWS distributed storage the code will be conveyed utilizing AWS CodeDeploy device [5]. It completely computerizes programming organizations, giving a quick and dependable technique for visit discharge plans.



Figure 2. Fully Integrated pipeline

Fig. 2 illustrates an application and its database integration within a CICD pipeline. Here, database migration is performed using Phinx migration tool, where up(), down(), and change() method will have major queries that manipulates the database. Coded migration file are going to be in project path under DB/migrations folder. Deployment pipeline should support deployments, during which either the database or the appliance changes.

5. Implementation

5.1 Continuous Integration

In Giltab, merge request are often wont to push the migration file, in order that Continuous Integration Pipeline will have file of the appliance with intended database changes. Once the merge request is approved, then the migration file are going to be moved to Continuous Deployment pipeline.

5.2 Continuous Delivery

In Continuous delivery pipeline, the migration file are going to be deployed in staging environment along side the appliance . After testing stage, the migration file are going to be deployed in staging environment. Fig. 3 represents the top to finish deployment, where it starts from code push into a central repository(Gitlab), build composer, testing and deployment [4]. Once it's integrated the merchandise delivered and monitoring takes place within the CICD Pipeline. This pipeline is an automatic system that executes a progressive set of test suites against the build [7]. Continuous Delivery is very automatable and in some cloud-computing environments, easily configurable



Figure 3. Continuous Integration and Continuous Delivery handled in database migration

6. Result

The implementation is tested using different requests on various databases of HRAPP application system. The results of this project shows that it takes less time for deploying data migrations using our system, in comparison to the prevailing method that uses shell scripts. With shell script deployment it takes 06.32 minutes, but in phinx migration deployment it takes 02.13 minutes for same merge request.

0 7 0 8	O D E Imps./gdab.com/sis/vu03/trapportine/pipetines/#0065627.builds			···· 日本	
😝 GitLab 🕬 Projects - 1	Grangs - Activity Miles	ones Seippets 🙆		🛛 🗸 Search or jump to 9. D	• n 🖙 e- 🌖
H hrappenline & Project D Repository	Merge brand	h 'develop' into 'mast	:er'		
Di haues (200 11 Merge Requests (3)	O Tplatifium suchar is a seniodas				
e CI/CD Hpelines	← ect20511	0			
Jubs	Pipeline Jaka (E				
Charts	Status 🕑 Bulld	0.6K	Name	Сангар	
≩ Operations □ Rephtry	() passed	#170468409	buildComposer	⊕ 00.01:19 ☐ 3 weeks ago	0
B Packages	© Staging				

Figure 4. Pipeline screen in Gitlab

Fig.4 illustrates deployment environments with time taken for every deployment. The tracking of entire migration process with time taken for deployment in Gitlab is presented. Integrating application and database in Gitlab with single pipeline enables delivery of software in few milliseconds with intended changes to application databases. the complete spectrum clearly shows issues, commits to merge request, reviews, builds, tests, deploys, deployment history, environments details and active pipelines

7. Conclusion

Fig.4 illustrates deployment environments with time taken for every deployment. The tracking of entire migration process with time taken for deployment in Gitlab is presented. Integrating application and database in Gitlab with single pipeline enables delivery of software in few milliseconds with intended changes to application databases. the complete spectrum clearly shows issues, commits to merge request, reviews, builds, tests, deploys, deployment history, environments details and active pipelines

References

- M. Artac, T. Borowiak, E. D. Nitto, M. Guerriero, "Model-driven continuous deployment for quality DevOps", *QUDOS@ISSTA*, pp. 40-41, 2016.
- [2] Lixian Xing, Yanhong Li, "Design and Application of Data Migration System in Heterogeneous Database", International Forum on Information Technology and Applications-2010.
- [3] M. Huttermann, "Beginning DevOps for Developers", DevOps for Developers, pp. 3-13, 2012
- [4] T. Lehtonen, S. Suonsyrjä, T. Kilamo, T. Mikkonen, "Defining metrics for continuous delivery and deployment pipeline", SPLST, pp. 16-30, 2015
- [5] Practicing Continuous Integration and Continuous Delivery on AWS, 2017, [online] Available: https://aws.amazon.com/
- [6] https://www.infoq.com/articles/deployment-pipeline-database-changes
- [7] V.D.Ambeth Kumar (2017), Efficient Routing for Low Rate Wireless Network a Novel Approach. International Journal of Image Mining, Vol. 2, Nos. 3/4, 2017, (2017)