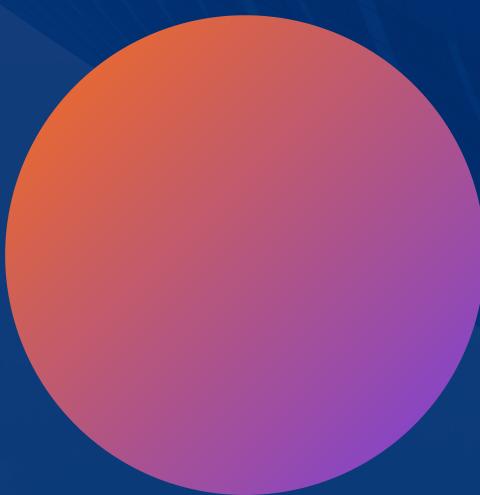


END TO END AUTOMATION ON CLOUD WITH BUILD PIPELINE



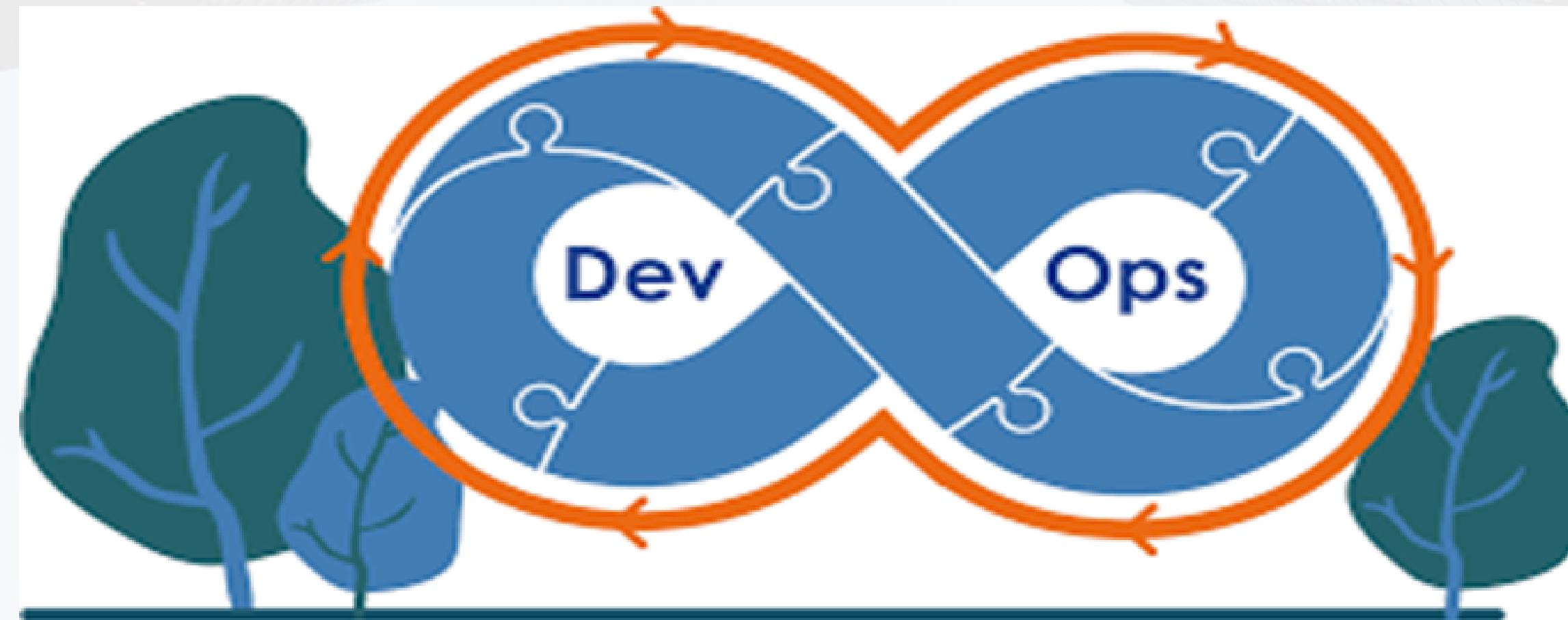
CONTENT

S.NO	CONTENT	PAGE NUMBER
1	Abstract	3
2	Existing System	4
3	Literature survey	5
4	Proposed System	8
5	System Architecture	9
6	Modules & Explanation	10
7	Screenshots	13
8	Conclusion	17
9	Future Enhancement	18



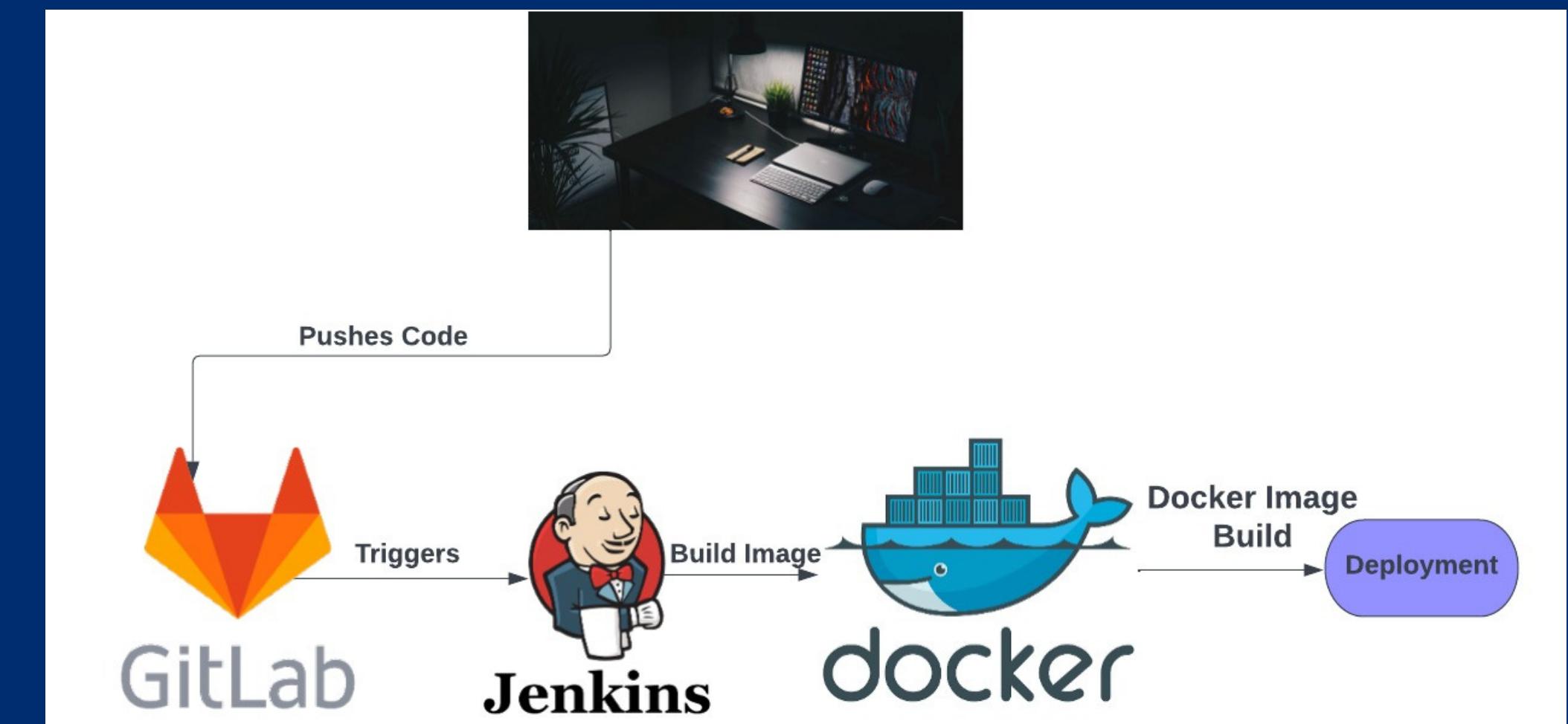
ABSTRACT

CI/CD Pipeline is used developing applications and delivering them across environments in an automated manner to improve performance and quality assurance. Continuous Integration (CI) and Continuous Delivery (CD) have emerged as a boon for traditional application development and release management practises, allowing them to deliver quality artefacts to customers on a continuous basis while receiving continuous integrated feedback. The goal of this paper is to develop a proof of concept for designing an effective framework for continuous integration, continuous testing, and continuous delivery that uses the build pipeline concept to automate source code compilation, code analysis, test execution, packaging, infrastructure provisioning, deployment, and notifications.



EXISTING SYSTEM:

CI IS A SEGMENT OF PRACTICES INVOLVED IN SOFTWARE PROGRAMMING PRINCIPLES. CI STATES THAT THE ENTIRE CODE FOR AN APPLICATION SHOULD BE KEPT IN A COMMON REPOSITORY SO THAT WHENEVER THE DEVELOPER CHECKS IN THE CODE INTO THE REPOSITORY, A SCRIPT IS TRIGGERED WHICH PICKS THE LATEST CODE FROM THE REPOSITORY, INTEGRATE IT WITH THE EXISTING CODE AND RUN THE TEST CASES DESIGNED ACCORDING TO THE APPLICATION. THERE ARE MULTIPLE CI (CONTINUOUS INTEGRATION) TOOLS AVAILABLE IN THE MARKET LIKE JENKINS, BAMBOO, GITLAB, SUBVERSION ETC.



LITERATURE SURVEY

S.NO	AUTHOR	TITLE	DEMERITS
1	Charanjot Singh, Nikita Seth Gaba, Manjot Kaur, Bhavleen Kaur.	Comparison of Different CI/CD Tools Integrated with Cloud Platform	It can be used only for a simpler, sophisticated and small microservice based project.
2	Qianying Liao.	Modelling CI/CD Pipeline Through Agent-Based Simulation	The budget is high, the investigation result is often influenced by the chosen technologies, platforms and hardware is more complicated
3	Aayush Agarwal, Subhash Gupta, Tanupriya Choudhury.	Continuous and Integrated Software Development using DevOps	This uses a long branching CI/CD pipeline which reduces efficiency and speed

LITERATURE SURVEY

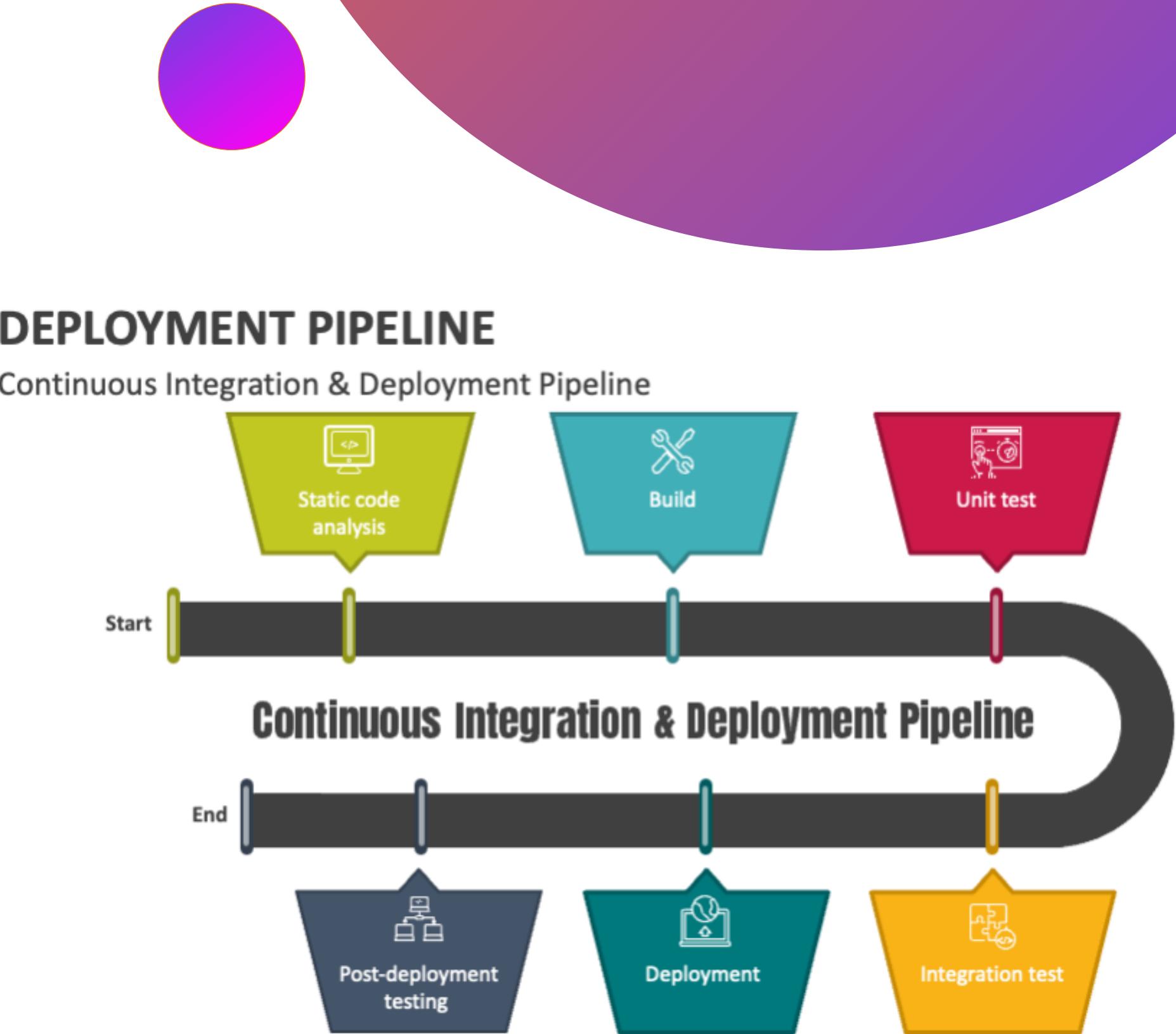
S.NO	AUTHOR	TITLE	DEMERITS
4	Bao-An Nguyen, Kuan-Yu Ho, Hsi-Min Chen.	End to End Automation On Cloud with Build Pipeline: The case for DevOps in Insurance Industry	The work needs to be focused on monitoring, security, quantifying benefits of end to end automation for continuous delivery.
5	Mansurali, P. Mary Jayanthi, R. Swamynathan, Tanupriya Choudhury	Continuous Delivery with Jenkins.	To make Jenkins able to fully embrace the CD revolution: indeed, the lack of best practices for versioning continuously is lacking in Jenkins.
6	Prashant Agrawal ,Neelam Rawat.	Devops, A New Approach To Cloud Development & Testing.	To Improve Collaboration: DevOps with cloud projects continuously emphasizes on building the bridge.
7	Thomas F. Düllmann; Christina Paule.	Exploiting DevOps Practices for Dependable and Secure Continuous Delivery Pipelines.	In the flow of securing and dependable of the content delivery the concentration to the pipe flow is not well organized.

LITERATURE SURVEY

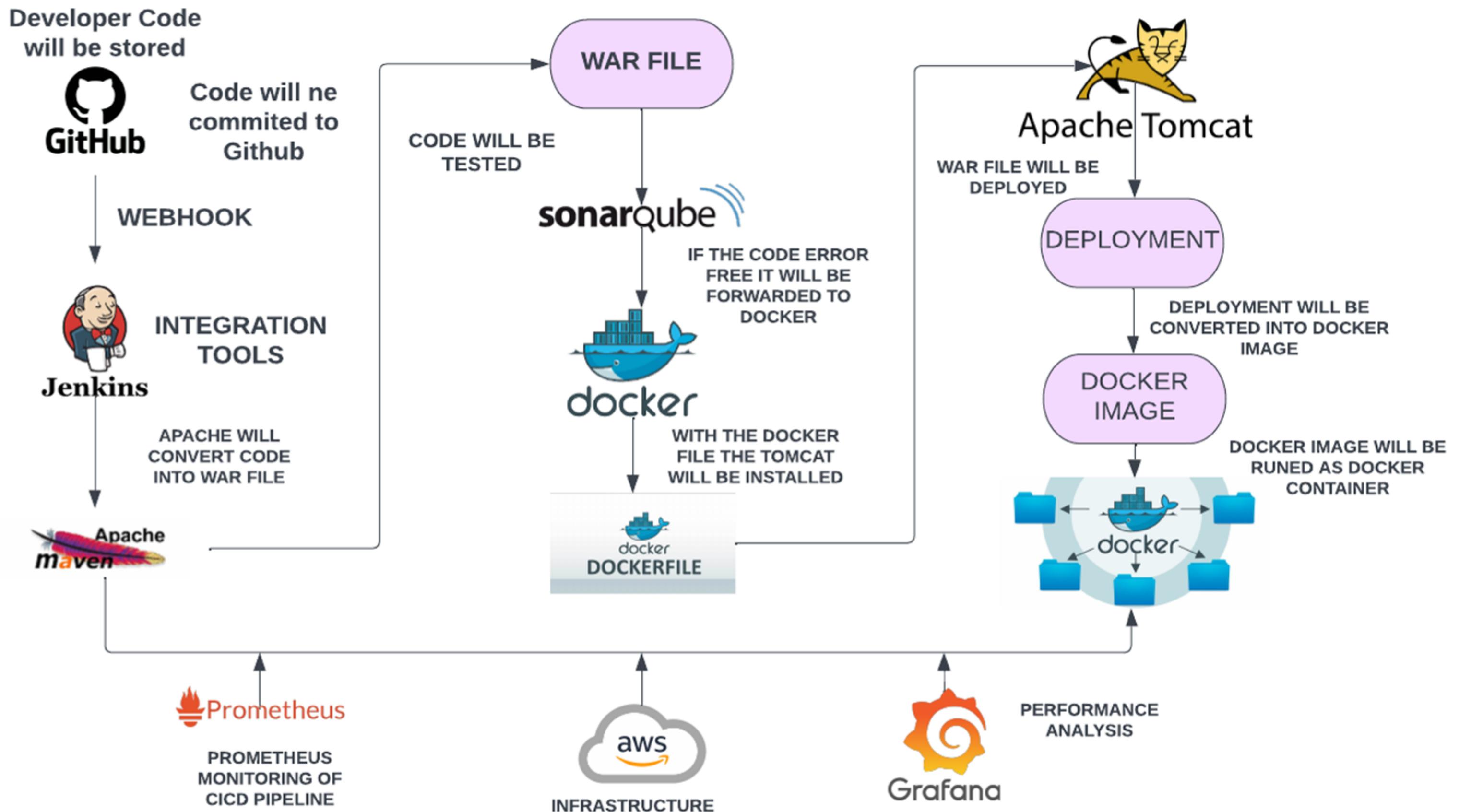
S.NO	AUTHOR	TITLE	DEMERITS
8	Jay Shah; ushyant Dubaria; John Widhalm.	A Survey of DevOps tools for Networking	This survey paper deals about the various devops tool which is out dated version and new tools is not discussed which is efficient
9	V. Arulkumar, R. Lathamanju	Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method	The flow of strat to finish is lengthy and steadoous process with less effieciency
10	Hasan Yasar	Sizing Exposed Credentials in GitHub Public Repositories for CI/CD	Automation is a central approach in continuous integration (CI) and continuous deployment (CD), but it does not come without new security risks. Automation often eliminates any human checks before building or deploying and, if it is not thoroughly planned it can result in major security issues such as exposing secret information inside configuration files, deployment scripts, or the code itself.

PROPOSED SYSTEM:

- The CI/CD Pipeline is a relative newcomer to the SDLC scene.
- It emerged from two trends: the application of Agile and Lean practices to operations work, and the general shift in business toward seeing the value of collaboration between development and operations staff at all stages of the SDLC process.
- The CI/CD Pipeline relies on effective tooling to help teams rapidly and reliably deploy and innovate for their customers.
- These tools automate manual tasks, help teams manage complex environments at scale, and keep engineers in control of the high velocity that is enabled by DevOps.



SYSTEM ARCHITECTURE:



BUILD STAGE:



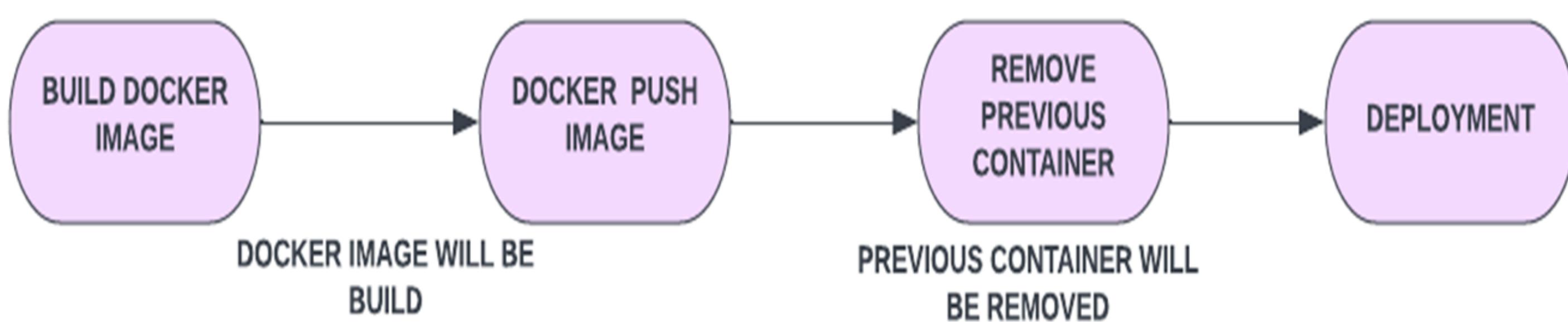
The developer develops the code and commits it in **Github** repository. Then the integration tool(**Jenkins**) integrates github with Jenkins. Then the code is pushed to **Maven Apache** which is build tool.it builds the code into packages and gives in WAR format.

TEST STAGE:



In this stage WAR file will be pushed to **Sonarqube** which is testing tool to check whether the code contains any Bugs, Vulnerabilities, Code Smell etc.. if the code is clear then it is pushed to docker. **Docker** install Apache Tomcat through docker file.

DEPLOYMENT STAGE:



Docker image will be build and that image will be pushed to the docker container. In Which the old container will be deleted in order to maintain last pushed container. Then it will be Deployed .

IMPLEMENTATION OF THE PROJECT

FOUR STAGES OF DEPLOYMENT BY DOCKER

nexus Questions and Exer... An architect's guide... CampBuzz Portal LII

Dashboard > myproj > Configure Delete Pipeline Full Stage View Open Blue Ocean Rename Pipeline Syntax GitHub Hook Log

Stage View

Average stage times:
(Average full run time: ~26s)

SCM Checkout	maven-buildstage	Build Docker Image	Docker Image Push	Remove Previous Container	Docker deployment
1s	6s	1s	15s	730ms	824ms
#55 May 11 14:55 1 commit	1s	6s	1s	15s	852ms
#54 May 11 14:53 1 commit	1s	6s	1s	15s	534ms
#53 May 11 14:40 1 commit	1s	6s	1s	16s	838ms
#52 May 11 14:38 1 commit	972ms	6s	1s	15s	540ms

Build History trend ▾

Filter builds... /

- #55 | May 11, 2023, 9:25 AM
- #54 | May 11, 2023, 9:23 AM
- #53 | May 11, 2023, 9:10 AM

PROMETHEUS INTEGRATION WITH JENKINS

 nexus  Questions and Exer...  An architect's guide...  CampBuzz Portal  LTI  LTI

 Prometheus Alerts Graph Status ▾ Help   

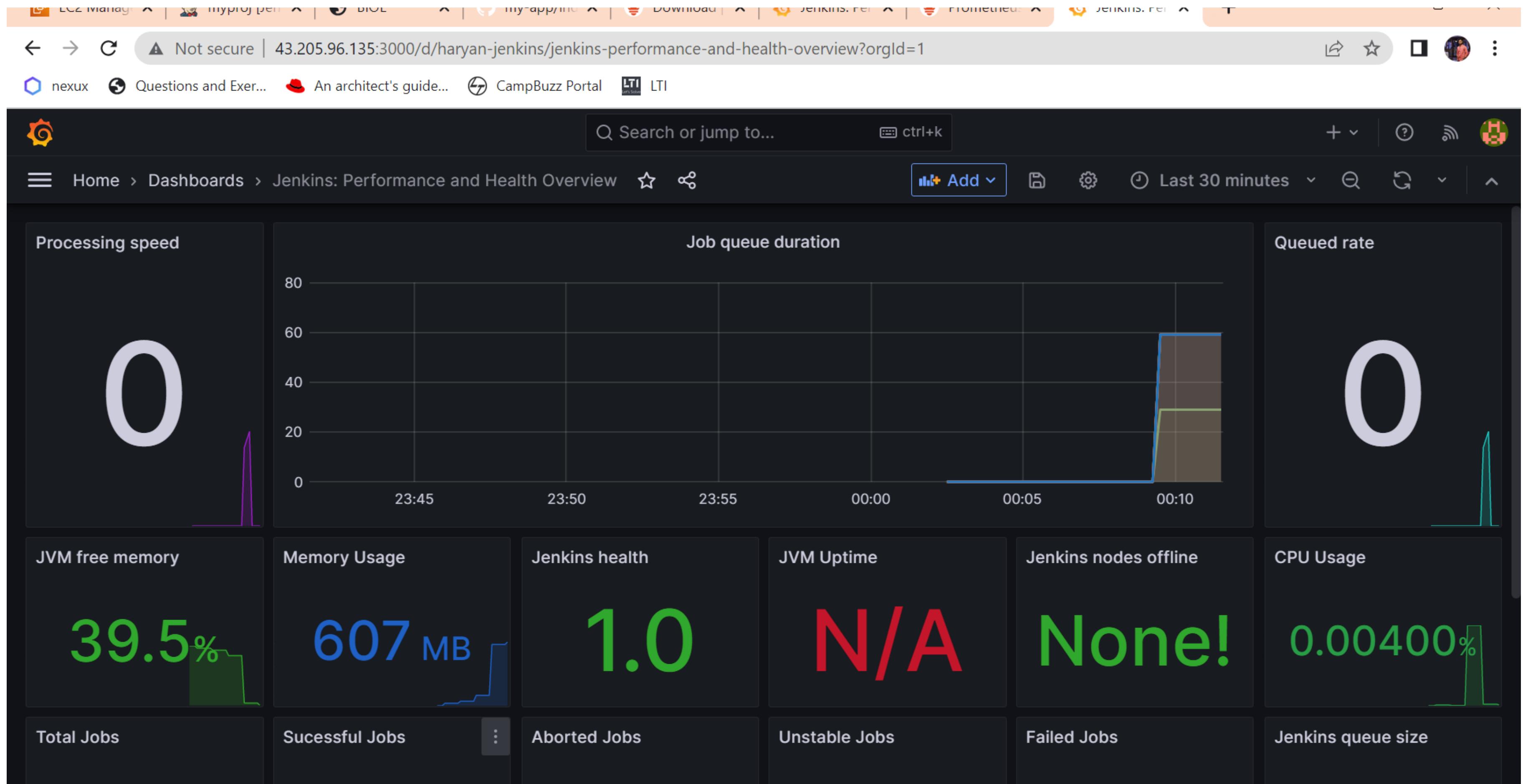
Targets

All scrape pools ▾ All Unhealthy Collapse All  Filter by endpoint or labels  Unknown  Unhealthy  Healthy

jenkins (1/1 up) 

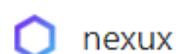
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://3.110.191.82:8080/prometheus	 UP	instance="3.110.191.82:8080" job="jenkins"	13.142s ago	8.701ms	

GRAFANA ANALYSIS CHART OF DIFFERENT MODULES

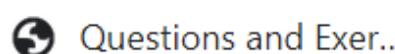


DEPLOYED WEBSITE

← → ⌂ Not secure | 3.110.191.82:8090/newapp/#services



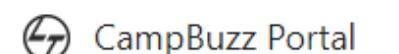
nexux



Questions and Exer...



An architect's guide...



CampBuzz Portal



LTI

SYPOSIYUM

TECHNICAL

NON-TECHNICAL

SPECIAL-EVENT

ABOUT US

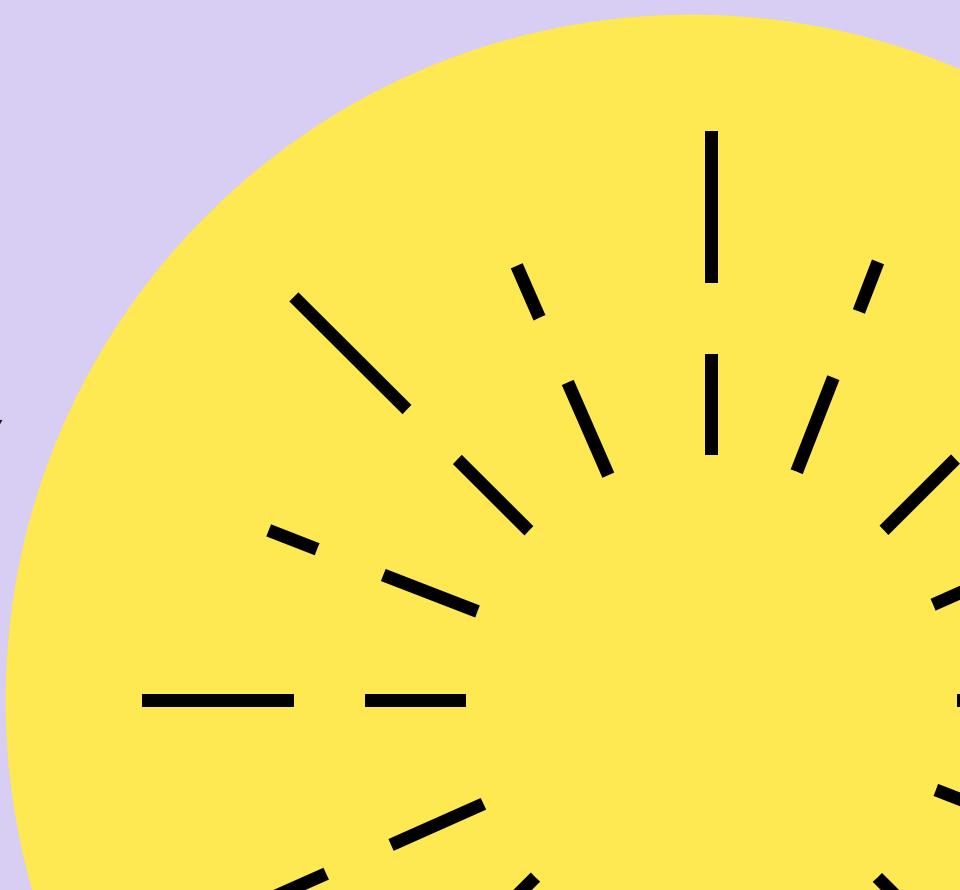
CONTACT

BETHLAHEM INSTITUTE OF ENGINEERING

CONCLUSION



CI/CD culture extends agile methodology by rapidly developing applications and automating their delivery across environments to improve performance and quality assurance. Continuous Integration (CI) and Continuous Delivery (CD) have emerged as boons to traditional application development and release management practises, allowing them to continuously deliver quality artefacts to customers while receiving continuous integrated feedback. The purpose of this paper is to create a proof of concept for creating an effective framework for continuous integration, continuous testing, and continuous delivery that employs the build pipeline concept to automate source code compilation, code analysis, test execution, packaging, infrastructure provisioning, deployment, and notifications.



FUTURE ENHANCEMENT

Now, the industry needs to streamline testing by reducing development tool complexity. Tools are generating ample data, but the next evolution is efficiently taking that data and using it to drive DevOps to even faster CI/CD cycles. The DevOps method consists many tools for the different methods like for integration there are Jenkins, Bamboo, Buddy. Apache Gump and for deployment ansible, chef, docker, Kubernetes. So, the future work will be used with the future tools with to be discovered for more efficiency, these can help improve the coordination and flow of information between different teams to adopt it to build, test, and deploy processes. With automation, DevOps will become even more agile and efficient. It will result in a better customer experience and higher-quality products. Automation will also make it easier for companies to scale their services globally. DevOps future relies on practical use cases in the area.



Thank You