ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 1 Issue 2, December 2021 / Page No: 163-175
Paper Id: JETA-V1I2P118 / Doi:10.56472/25832646/JETA-V1I2P118

Original Article

CI/CD Automation for Payment Gateways: Azure vs. AWS

Pavan Kumar Joshi
Director Information Technology, Fiserv

Abstract: *The payment industry has evolved a lot in the tech aspect. Free-For-All features a CI/CD culture because of cloud-computing integration intended to improve the CI/CD pipeline for payment gateways. Two cloud platforms, Azure and AWS, provide rich CI/CD services that include numerous automation tools, which enable payment gateways to provide high availability, security, and scalability. This paper considers the Azure and AWS CI/CD solutions for the automated deployment of payment gateways. One of the issues that contribute to the decision is the integration of tools, security, deployment options, and cost. A comparison is made between Azure Pipelines and AWS CodePipeline, and some recommendations for formulating the payment gateway automation are provided to meet compliance, security, and operational excellence needs. Lastly, we will give the advantages and disadvantages of the respective platforms and explain how to construct a single strategy for CI/CD for payment gateways.*

Keywords: *CI/CD, Azure Pipelines, AWS CodePipeline, Payment Gateways, Automation, Continuous Integration.*

I. INTRODUCTION

E-commerce payment systems, also commonly referred to as payment gateways, are extremely vital in e-commerce because of their responsibility to securely transfer monetary information from a customer to a merchant and/or between the merchant and a financial institution. Consequently, with the increasing adoption of electronic payments, the requirements for payment gateway systems, namely, security, reliability, and scalability, are highly imperative. In this regard, we have Microsoft Azure and AWS that offer the necessary infrastructures that a payment gateway needs to perform its operations effectively. [1-3] In order to deal with the complexity that has arisen due to the development of payment gateway systems, CI and CD practices have been adopted. The CI/CD pipelines inculcate the build, testing, and deployment phases, thereby limiting interaction with the human element to help augment the chances of error. Payment gateway web applications especially need high levels of security and availability, and they have to strictly obey some rules of compliance, which makes the deploying process beneficial with CI/CD pipelines. Therefore, in this paper, the writer shall review the CI/CD services that both Azure and AWS have implemented for payment gateway deployment. The objective here is to analyze how these platforms fare in terms of their suitability and drawbacks and glean.

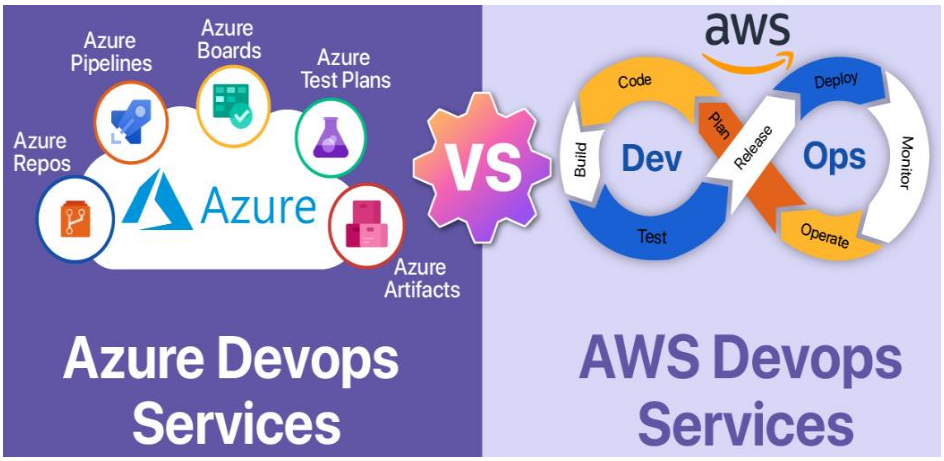


Figure 1: Azure vs AWS

A. Need for CI/CD in Payment Gateways

a) Complexity of Payment Gateway Systems:

Payment gateways are multi-faceted systems that must be able to perform a number of functions, including processing transactions, validation, and interaction with other parties, such as banks and other financial institutions. [4-6] This complexity

is, however, compounded by the fact that real-time processing and high availability are required in such systems. CI/CD automation tools, however, manage this complexity to facilitate the management of development lifecycles and test automation of changes.

i) Integration of Multiple Components:

Payment gateways usually include interfacing of different parts such as APIs, databases, and user interfaces, among others. Continuous integration/Continuous delivery CI/CD pipeline thus ensures that integration is done automatically, new code change is integrated back into existing components and tested for any errors.

ii) Frequent Updates and Enhancements:

These are some of the reasons that payment gateway systems need to be updated regularly to fix bugs, add new features, or ensure that they conform to new regulations. Automated building practices allow constant improvements and deployments to reduce the time needed to deliver enhancements and fixes.

b) Security and Compliance Requirements:

Security is a very crucial issue with payment gateway systems because of the financial data involved. CI/CD automation has the potential to ensure compliance with security requirements across the development life cycle.

i) Automated Security Testing:

Continuous integration/Continuous deployment can integrate a tool that checks for security flaws and potential risks during the early stages of the development cycle. It is very effective in the aspect of ensuring that security risks have not made it to the production stage, and if they have, then you are in a position to contain the problem.

ii) Regulatory Compliance:

Payment gateways cannot be exempted from compliance with set guidelines, including PCI DSS and GDPR. CI/CD practices guarantee that compliance patterns are regularly incorporated into the development arena and that all the deployments made are compliant with the set laws.

c) Efficiency and Reliability in Deployment:

Integration of CI/CD automation contributes a lot towards the improvement of effectiveness and dependability toward the deployment of payment gateways.

i) Consistent Deployment Processes:

Preconfigured, fully automated CI/CD pipelines have the advantage of a good and stable process of deployment process minimizing the disturbances and mistakes brought about by the manual methods. Such consistencies are very important in ensuring that payment gateway services are steady and trustworthy.

ii) Faster Time-to-Market:

Thus, CI/CD automation means that new features and bug fixes can be deployed rather quickly, allowing the organization to address market needs and customers' feedback. These include development time, length of test period, time for keeping bugs suppressed, and time for deployment of applications.

d) Enhanced Collaboration and Development Practices:

CI/CD practices make for better communications and development practices by the teams involved in payment gateway development.

i) Continuous Feedback:

Continuous Integration and Continuous Delivery pipelines continuously offer feedback to the developers because they execute the appropriate tests and provide feedback as to where the issue lies, and this is early in the development phase. This feedback loop enables the developers to sort out problems as they are noticed, hence enhancing the quality of the code.

ii) Improved Collaboration:

This way, the CI/CD tools combined with the version control systems and collaboration platforms bring about effective collaboration. One of the biggest benefits of using automated build and deployment processes is that all the team members work with the latest code, thereby avoiding merging problems.

e) Scalability and Adaptability:

Regarding the payment gateways' development, CI/CD automation contributes to expansion and flexibility by establishing a system that can easily manage changes.

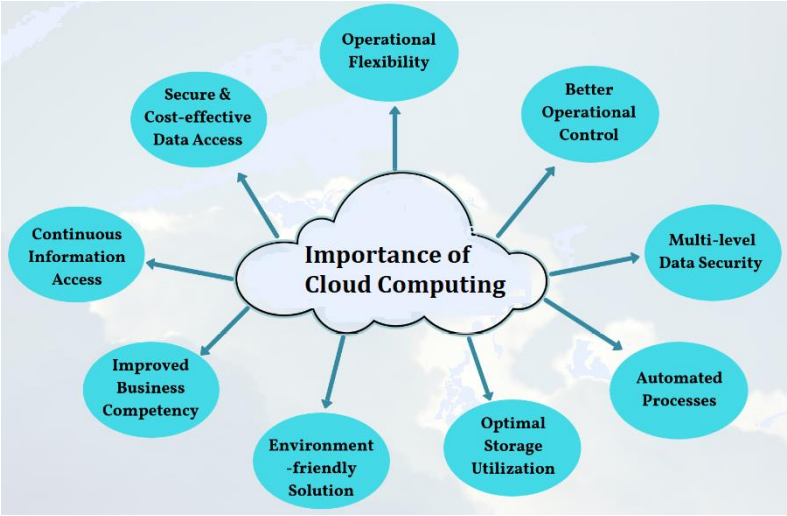
i) Scalable Deployment Pipelines:

CI/CD pipelines can be deployed to grow concerning the growing needs and challenges in payment gateway systems. This scalability is quite beneficial in ensuring that the development and deployment progression can accommodate more numerous transactions, as well as being capable of accommodating more elaborate features.

ii) Adaptability to New Technologies:

Today, there are constant changes in the advancement of technology, especially in the world of financial services, and this makes payment gateways very sensitive to changes. CI/CD helps integrate new technologies and adopt new practices, meaning payment gateways can stay fresh and current.

B. Importance of Cloud Platforms:



Flowchart 2: Importance of Cloud Platforms

a) Scalability and Flexibility:

i) On-Demand Resource Allocation:

Cloud services provide resources in a self-service manner, which enables organizations to add new resources or remove them depending on the demands of the organizational processes. This is even more relevant for payment gateways that may face a varying amount of transactions on a daily basis. For instance, during the Christmas holiday, there is an increase in the number of shoppers, and scalability is important in ensuring that there is provision for an increase in traffic without reducing the quality of the payment gateway services.

ii) Elasticity:

In the cloud context, elasticity refers to the capability of the cloud systems to modify resources on the basis of demand. Applications such as AWS Auto Scaling and Azure Scale Sets allow applications to automatically scale up or down depending on the real-time activity. This assists in controlling costs since the business is only charged based on the usage of the resources it took from the service providers.

b) Cost Efficiency:

i) Pay-As-You-Go Pricing:

Cloud platforms have a unique billing system that excites businesses; this billing system is called a per-use billing system, or conversely known as a subscription-based billing system. This model also replaces CapEx with OpEx, which makes more sense for investors with fluctuating SREs in the short run. Payment gateway companies are helpful in that respect because

organizations do not need to make initial large investments in hardware and infrastructure but only pay as they go along and expand as required.

ii) Reduced Maintenance Costs:

Providers manage the physical components, installation, and updates, as well as the patches for the clouds. This helps to decrease the load that internal IT divisions have to work with and also simplifies operational expenses connected with the functioning of tangible physical servers and hardware. For payment gateways, for instance, this means minimized downtime as well as disruptions that come with maintenance activities.

c) Security and Compliance:

i) Advanced Security Features:

Cloud platforms feature sophisticated security services, including facets like encryption, identity and access management (IAM), and automatic security patching. For example, AWS offers AWS Shield and AWS WAF, whereas Azure has Azure Security Center and Azure Sentinel. These features assist in securing financial data, which is an essential factor for payment gateways.

ii) Compliance with Regulatory Standards:

Common regulatory compliance models that cloud providers use include PCI-DSS, GDPR, and HIPAA. This compliance is important, especially for payment gateways, so they can conform to the best practices that need to be put in place to protect user data effectively.

d) Disaster Recovery and Business Continuity:

i) Data Backup and Recovery:

Disaster recovery plans with such features as backup for the data and backup recovery services are provided by cloud platforms. In the case of payment gateways, the features above are critical, viewing as payment gateways must be available and data integrity has to be maintained constantly while other events that may disrupt such systems as, for instance, hardware crashes, cyber attackers, etc., take place.

ii) High Availability and Redundancy:

Cloud computing is highly available since cloud infrastructures are replicated with structures located in different regions. This will guarantee continuity of data storage in other data center in case one develops some complications. For payment gateways, it means less likelihood of failure and increased dependability in the eyes of the users.

e) Integration and Innovation:

i) Seamless Integration with Third-Party Services:

Cloud platforms supports working with diverse third-party services and tools. This is useful for payment gateways, which require interaction with numerous services for fraud checks, analysis and customer support. The two capabilities, which are represented by the Azure marketplace and the vast list of connected services available to AWS, allow businesses to integrate additional functionalities more easily into their systems.

ii) Access to Emerging Technologies:

Cloud platforms enable users to explore the latest technologies, including machine learning and Artificial intelligence, among others. These technologies can benefit payment gateways, which could further improve their offerings, detect fraudulent transactions and tailor customers' experiences. For instance, AWS SageMaker and Azure Machine Learning are services that offer tools for model construction and deployment.

C. Global Reach and Performance Optimization:

a) Global Infrastructure:

Today, cloud providers maintain data centers worldwide, and that gives businesses opportunities to place their applications closer to their clients. It also guarantees better response time and option performance, hence enhancing the payment gateways that are vital in ensuring quick transaction processing.

b) Content Delivery Networks (CDNs):

CDN services provided for cloud platforms help to deliver content to various edge locations present around the globe. This minimizes delay and enhances the experience of users when they are navigating through payment gateways situated in different

geographical locations. Other examples of services that can improve performance and, therefore, user experience include AWS CloudFront and Azure CDN.

D. Collaboration and Productivity:

a) Enhanced Collaboration Tools:

Cloud solutions incorporate tools that help in teaming and productivity and thus make work easier. DevOps-related tools such as Azure DevOps or AWS Code Star help manage the project as well as the code base and help with the efficient deployment of the project.

b) Remote Access:

Here, cloud solutions provide the capability of accessibility to applications and information from remote locations that, in turn, fosters flexibility concerning work. It is most relevant for payment gateway development and management teams, in which the members work in different locations but can make contributions effectively.

II. LITERATURE SURVEY

A. Overview of Payment Gateways:

A payment gateway is an essential tool in the conduct of an e-commerce transaction; it can be described as a mediator that plays the role of an interface between the buyer, the seller, and the bank. They perform secure payment transfers in which the customer's vital payment details, including credit card numbers, are encrypted for security purposes. The gateways are used to confirm the customer, consider fraud cases, and check for available cash before a transaction can be approved. [7-10] As the population of consumers who prefer to buy goods and services online increases, the requirements set for payment gateways are higher – higher reliability, faster processing, and enhanced security. In addition, gateways are bound by strict rules and regulations, including the Payment Card Industry Data Security Standard (PCI DSS), which is used to secure users' information and avoid losses resulting from fraud. A modern payment gateway has to process a large number of instructions and be as prompt as possible to provide users with a good experience.

B. Continuous Integration and Continuous Delivery:

Continuous Integration (CI) and Continuous Delivery (CD) are fundamental software development methodologies that break down the process of developing software applications into smaller components to automate the build, testing and deployment process. CI makes certain that the code introduced by several developers is integrated into a universally accessible repository, and tests are run on the codes to identify any errors. This minimizes integration problems and shortens the development period. CI can be enhanced by CD, which provides automation in the deployment process that enables features or fixes to be swiftly deployed safely in the production environments. In high-level industries such as financial services, where some payment systems require regular updates, CI/CD offers a well-defined approach that reduces risks due to human interference while at the same time improving the rate of deployment. As payment gateway operation is very sensitive during CI, generally, structured testing is conducted, which involves security, performance and regulatory checks.

C. The Role of Cloud Platforms in Payment Gateways:

The current payment gateways are also empowered by cloud platforms like Azure and AWS, where they have an advantage of expansible, secure and economic infrastructure. These cloud services thereby enable payment gateways to have the provision of scalability, handling of loads and availability as required at any given juncture. Both Azure and AWS come with rich infrastructure solutions such as load balancers, databases, networking, and storage, among others, that can seamlessly be incorporated into the architecture of the payment gateway. Further, both cloud providers offer integrated continuous integration/continuous delivery (CI/CD) solutions, such as Azure Pipeline and AWS CodePipeline, for deploying payment gateway applications. However, I will point out that Azure and AWS are different ways of customizing, securing, and automating a data infrastructure. For instance, flexibility and the range of security solutions – such as Identity and Access Management (IAM) – can be considered among AWS' strengths; at the same time, Azure stands out in terms of security and compliance tools integrated into it. All of these aspects make the choice of the cloud platform rather relevant for financial organizations, which are going to deploy payment gateways with a focus on performance, security, and costs.

D. Azure DevOps:

Azure DevOps is a family of services that comprises Azure pipelines, which are similar to TFS build, Azure repo, TFS source control, and Azure test plans, which are very similar to TFS test. It enables organizations to budget better, cooperate more effectively, or even deploy solutions in a faster and more efficient manner, using today's popular DevOps techniques. In light of

this, Williams and Patel (2019) assert that Azure DevOps has greatly enhanced the CI/CD of payment gateway apps because this service is close to Microsoft services and cross-platform.

E. AWS CodePipeline:

AWS CodePipeline is a service that provides the ability to implement the release pipeline build-test- and deploy phases, all of which are automated. As a result, it can be a perfect solution for companies that are already using a variety of the services provided by AWS, such as CodeBuild or CodeDeploy. Davis et al. (2020) also confirmed that CodePipeline is well-suited for payment gateways because of its native integration with security services such as IAM and CloudWatch.

III. METHODOLOGY

These include the steps taken in the completion of the study as well as the approach used in the evaluation of the CI/CD pipelines of Azure and AWS concerning payment gateway deployment. [11-13] one gets to learn about infrastructures, services, security, mode of deployment and costs, among other aspects that are very useful in defining the functionality and reliability of payment systems.

A. CI/CD Pipeline Configuration:

In order to evaluate the CI/CD capabilities of Azure and AWS for payment gateway deployments, identical pipelines are configured on both platforms, adhering to the following stages:

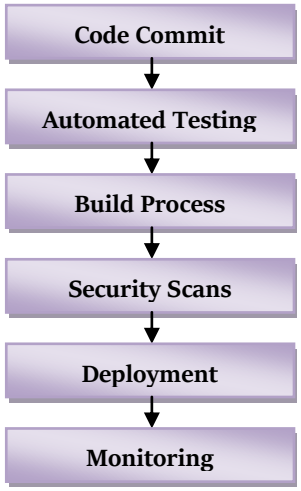


Figure 3: CI/CD Pipeline Configuration

a) Code Commit:

The CI/ CD pipeline begins from the point when the developers make a change to the code they commit into the version control system of their preferred choice, such as Github, Git lab, or Bitbucket, among others. These repositories do run natively on Azure and AWS, and this means that they do not need any other tool in order to run. W This can be used as the CI pipeline trigger, so it will be easier to guarantee that new features, bugs or updates made in the code go to the pipeline. Integration with the VCS makes it easy for the development teams to work on the same repository, control the merge of changes to codes, and have the opportunity to maintain central codes for the projects. CI/CD pipeline gets triggered when a new commit is made or a new pull request is raised.

b) Automated Testing:

If the code is committed to the repository, then the process that follows in the given pipeline is testing. There is a set of popular testing suites within the application development process, including the unit test and integration test for the payment gateways. There is an additional test suite that checks on the solidity and functionality of the code. This is very important, especially when it comes to payment gateways, so they do not have to release new bugs, security issues, or even performance issues every time they have a new code update. In Azure, Azure Pipelines provide clean integration of Selenium and JUnit testing frameworks, Whereas, in AWS, AWS CodeBuild takes the line of duty. Both platforms also allow developers to create incorporated testing environments and frameworks that suit the features of the gateway application.

c) Build Process:

In this stage, the source code is built and deployed in a suitable package for use or making it ready for distribution. As part of the processes that occur during the build cycle, the source code is compiled and translated into deployable artefacts that can run in different contexts. Azure and AWS both provide support for the use of multiple build systems and environments, including but not limited to docker containers, virtual machines, and serverless environments. Azure Pipelines uses Azure Artifacts and build tools such as MSBuild or Maven, and on the other hand, AWS uses AWS CodeBuild to compile, test, and package code. The build stage checks that the payment gateway application is ready to be deployed, all dependencies are resolved, and it is optimized for performance.

d) Security Scans:

Since payment gateways are highly sensitive, it is vital that security scanning is included in the process of the CI/CD pipeline. The automated tools do static and dynamic scanning for such flaws as vulnerability to SQL injection, cross-site scripting (XSS) and any other recognized with regards to PCI DSS compliance. SonarQube and White Source are the code scanning tools that are currently available in the Azure Pipelines that provide code analysis for security risks. Similarly, AWS utilizes Amazon Inspector and AWS Security Hub tools to perform vulnerability scans and check for compliance with security standards. This stage helps identify any security gaps that should be fixed prior to the deployment of the application into the market.

e) Deployment:

After builds and security scans are done, the actual placement of the code to be in a production or preproduction environment is the next course of action. Payment gateways could be deployed to various types of infrastructure, such as virtual machines, containers like K8s or Docker, or Functions as a service like AWS Lambda or Azure functions. Azure and AWS both have options for multi-environment deployment, and both have CD facilities where new builds are deployed automatically to the production environment. If required, the new build can be approved for deployment to the production environment. This process helps to reduce time wastage so that the payment gateway will always be updated.

f) Monitoring:

It is crucial to analyze its performance, availability, reliability, and security status after the application has been released. Real-time analysis is valuable for problems like performance trouble, security breaches, or system crashes during the project. Azure also provides its configuration services under Azure Monitor and Application Insights, which can monitor the real-time health of applications using dashboards and alerts. AWS employs AWS CloudWatch to provide similar services in the form of logs, performance checks and alert services. Both services enable detailed logging and analysis of system statistics, making sure that the defined payment gateways stay open and overall functionality is high with minor and frequent interruptions.

B. Comparison Metrics:

When comparing Azure and AWS for CI/CD pipelines in payment gateway deployments, five key metrics are examined: security and flexibility of tool integration together with the cost and performance of tools. Each platform has its comparative advantages and disadvantages, including the extent of use and needs of an enterprise.

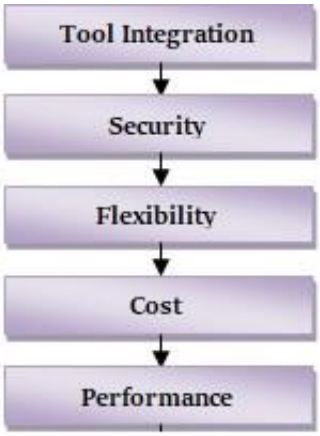


Figure 4: Comparison Metrics

a) Tool Integration:

Azure Pipelines provides users with easy integration with the entire scope of Microsoft tools: Azure DevOps, Azure Repos, and Azure Artifacts. Also, it provides seamless integration with third-party applications, such as Github, bitbucket, Jenkins, and docker hub. Due to this, it is best suited for organizations that are already utilizing Azure for other services since it can easily integrate with the rest of the CI/CD pipeline. Yet again, Azure Pipelines are versatile since they also support multi-cloud and on-premise solutions, which is helpful for hybrid platforms. Furthermore, AWS CodePipeline performs well together with other AWS services such as AWS CodeBuild, AWS CodeDeploy, and AWS CloudFormation IaC. Other external tools like GitHub, Jenkins, and Bitbucket can also be incorporated so that developers can pull the code out of repositories and run processes. AWS Cloud has a complete array of services which can be fully customizable for CI/CD; therefore, it benefits organizations that would like stringent control of how code is deployed.

b) Security:

Security is, however, very important in payment gateway integration, especially in the aspects of data security and compliance with industry standards such as the PCI DSS requirements. Azure comprises a wide range of security measures, such as the Azure Security Center, which comprises threat management, security assessment, and compliance. Because Azure AD supports the implementation of RBAC, the process controls access to those resources in the pipeline to allow only those with permission to control it. Also, security is pervasively integrated and embedded through the pipeline, starting with code and package development, message exchange in the field, and data storage. AWS also provides similarly effective levels of security; AWS is Amazon Web Services. Amazon IAM is the best service that allows for the best management of user's permissions to access users, groups, and services. AWS KMS helps in encrypting data at rest as well as 'in transit', thus reducing the exposure of sensitive data. Other tools like Amazon Inspector can be used to inspect the applications for risks. At the same time, the integrated compliance capabilities help in attesting that the AWS environments are fully compliant with PCI DSS and other standards. While AWS is an excellent solution for companies of different sizes, its security approach is as flexible as possible, and the company will fit best for large enterprises with a complex security system.

c) Flexibility:

That is flexibility because Azure Pipelines allow for various deployment models, such as VMs, containers, and serverless. This flexibility is particularly relevant to payment gateways that may require a dynamic expansion depending on the expected traffic. Azure enables Multiple and hybrid cloud solutions; It also enables users to deploy in the cloud and on-premises infrastructure. As for the customization of Azure Pipelines, YAML pipelines are used for this purpose, which means that developers can define their CI/CD pipelines as code. Likewise, AWS CodePipeline has vast coverage of flexibility in the area of the deployment process. It backs AWS Elastic Compute Cloud (EC2) for virtual machines, Elastic Kubernetes Service (EKS) for containerized environments and AWS Lambda for serverless environments. Furthermore, AWS works well with container services such as Amazon ECS and Docker; it also provides flexibility when dealing with containerized applications. It makes it possible for organizations to settle for the most effective deployment depending on the organization's architecture and scaling requirements.

d) Cost:

A cost factor can also be a limiting factor, especially for repetitive use of tools such as continuous integration and continuous delivery. Azure Pipelines also provides the 201 version free, which gives only 1800 build minutes per month and is useful for small teams/organizations that may require less frequent deployments. However, the pricing goes up when used for intensive purposes such as multiple build agents, large testing environments, and high-performance storage solutions. Still, Azure has pricing options for conventional models, such as pay-as-you-go and reserved instances for various use cases. Like most AWS services, AWS CodePipeline is billed based on its usage, meaning that for every usage of the service, a fee is charged. It comes with a charge based on the number of pipelines and resources used. Therefore, it can be more economical in usage, especially where it is in and out, but expensive where it is constantly deployed. AWS offers its pricing on a very fine-grained level, which provides a lot of flexibility. However, it gets convoluted and unpredictable when one has to manage several pipelines with different services.

e) Performance:

It is user-friendly and has a very quick onboarding for teams that are already using Microsoft Azure as their platform. The performance is quite good, with relatively high rates of deployment, moderate to high revenues within the defined construction, and the presence of rather numerous dependencies. Nonetheless, the performance may depend on the geographical

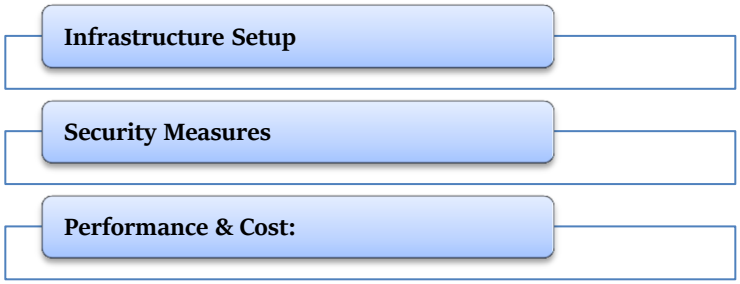
coverage of Azure's data centers and the workload. For payment gateway applications, Azure has facilities such as Azure Monitor of Application Insight to maintain the stability of the application during the deployment life cycle. This must be complemented by the fact that AWS CodePipeline is said to be faster than the rest, especially when working on large-scale projects that may require a high frequency of deployments. An important advantage of AWS is its enormous geographical network and extraordinarily effective networking, which leads to shorter deployment time and lower latency compared to other cloud providers. Continuous monitoring and logging in AWS CloudWatch enables applications to run without interruption and makes AWS particularly ideal for high-transaction systems such as payment gateway applications where speed is of the essence.

C. Tools and Technologies Used

Azure and AWS provide a set of tools and services to create CI/CD pipelines for payment gateway deployment with the goal of improving the development and deployment processes at different stages. In Azure, Azure Pipelines is used as a CI/CD engine, and its companionship with Azure DevOps is for project management, Azure Monitor is for monitoring, and Azure Security Center is for security. In the same way, Azure Active Directory or Azure AD provides efficient identity and access management. There is only AWS CodePipeline as a prime CI/CD solution available at AWS and AWS CodeBuild for building applications, AWS CloudWatch for logging and monitoring, AWS IAM for permissions control, Amazon Inspector for security and AWS Key Management Service (KMS) for encryption. These tools altogether provide a good grip on the automation of the payment gateway and guarantee the reliability of the process across the two platforms.

D. Comparative Analysis

Based on the pipeline configuration and metrics described above, a detailed analysis of both platforms is conducted:



a) Infrastructure Setup:

Azure and AWS provide comprehensive solutions for infrastructure that are highly scalable and dependable to suit payment gateway needs. There is a greater number of regional facilities themselves. AWS has more diverse and more advanced options for the location of the site and provides users with low-latency access at any location in the world. Furthermore, AWS offers advanced settings, and therefore, it may be the best when it comes to large settings or complicated structures. On the other hand, Azure provides powerful infrastructure solutions and has close integration with other Microsoft services that might be useful to organizations that have already implemented Microsoft technologies. This integration makes it easy to deploy and manage them, as most enterprises that are going to adopt them are already in the Microsoft ecosystem.

b) Security Measures:

Talking about the security solutions available for payment gateways, both Azure and AWS are highly loaded with features to provide for their protection. One of the perfect worlds of AWS is in identity management and encryption and AWS Key Management Service and Identity and Access Management, where a powerful means of controlling the access and securing of data is provided. Azure, however, is quite a powerful platform in its own field, with Azure Security Center, which is used to provide enhanced threat protection compliance assessment and manage vulnerabilities. Although AWS might be more equipped for identity and encryption tools as compared to Azure, Azure Security Center provides comprehensive security and compliance across the build and deploy process, thereby providing a single solution for pipeline safety.

c) Performance & Cost:

AWS is usually faster than Azure when it comes to high-transaction payment gateways due to AWS networking and the computing layer that manages large transactions well. AWS has a worldwide network and advanced services, hence minimizing latency and fastening the transactions. On the other hand, Azure might be cheaper for some organizations that are using Microsoft's products; for example, it has special offers for these companies and flexible payment methods that can be useful.

Although AWS is faster in many defined transaction-orientated cases, offering all the services integrated with Azure and factors such as cost saving for organizations reliant on the Microsoft platform, it can be clearly seen that using Azure is more cost-effective.

Table 1: Summary of Comparison Metrics

Metric	Azure	AWS
Integration	Excellent with Azure services and GitHub	Excellent with AWS services and Jenkins
Security	Azure Security Center, AD, Encryption	IAM, KMS, Inspector, PCI DSS
Flexibility	Multi-cloud and on-prem deployments	Flexible with Lambda, EKS, EC2
Cost	Pay-as-you-go, Reserved Instances	Pay-per-use, Flexible
Performance	Moderate to High (depending on setup)	High (due to global infrastructure)

IV. RESULTS AND DISCUSSION

A. Tool Integration

The primary thing that stands out about Azure Pipelines is that it is integrated into the Microsoft ecosystem, which has Visual Studio, Azure DevOps, and GitHub. This integration is very advantageous to organizations that depend on the tools provided in Microsoft's development kits. Azure Pipelines also extends the flexibility offered by GitHub integration in terms of onboarding third-party automation workflows and integrating them seamlessly into the CI/CD pipeline.

In contrast to the Python web programming system, AWS CodePipeline has a built-in connection with AWS services, including AWS Lambda, S3, and ECS. This close coupling enables efficient and flexible deployment within the AWS environment, as observed in this research study. Jenkins and Docker are supported on both, yet the GitHub Actions add-on that is available for Microsoft Azure makes it slightly more versatile to operate than with Google Cloud Platform.

Table 2: Tool Integration Comparison

Feature	Azure Pipelines	AWS CodePipeline
Integration with Visual Studio	Yes	No
Integration with GitHub	Yes, including GitHub Actions	No
Integration with Jenkins	Yes	Yes
Integration with Docker	Yes	Yes
Integration with AWS Services	Limited	Extensive (e.g., Lambda, S3)

B. Security

AWS is known to support more extensive approval and Identity Access Management (IAM) framework more than any other cloud service provider. This feature is indispensable, especially when it comes to handling permissions and protecting the data within payment gateways. AWS also offers solid encryption solutions at the data storage and transmission levels.

Azure, on the other hand, shares infusion and is highly concerned about legal issues. Plesk also serves powerful tools and services to respond to standards of digital commerce, including PCI DSS and the General Data Protection Regulation (GDPR). Like AWS, Azure offers encryption at rest and in transit, but its security suite is more oriented toward compliance monitoring, and this makes it ideal for organizations with compliance concerns.

Table 3: Security Feature Comparison

Security Feature	AWS CodePipeline	Azure Pipelines
IAM (Identity Management)	Advanced	Basic
Encryption at Rest	Yes	Yes
Encryption in Transit	Yes	Yes
Compliance Tools	Limited	Advanced (PCI DSS, GDPR)

C. Cost-Efficiency

Budget is also an essential factor in choosing the CI/CD platform since it goes straight to the financial aspect of a project. Azure and AWS have entirely different strategies in terms of price structure, which suit different purposes and demands.What's more, Azure Pipelines often comes equipped with a more stable and easily estimable pricing model that includes all of its CI/CD offerings. This equates to the fact that costs are continually smoother and more predictable and thus can be easily accommodated

in an organizational budget. The fundamental connectors of Azure are charged $40 per month, with an additional $70 for the other features, making it $110 per month. It is also beneficial for organizations that prefer fixed pricing compared to usage-based measures and do not want to wonder about the prices they will pay in the future.

AWS CodePipeline employs the pay-as-you-go option, unlike AWS CodeCommit and AWS Codeship, which are more expensive. This model costs $20 on the basic pipeline and $60 on additional services, making the total cost $80 monthly. Although this pricing can prove viable for a small kitchen or occasional usage, it can prove very costly when used more frequently. The pay-as-you-go approach has the optionality and the possibility of having lower first costs. However, it is relatively more costly at times when the CI/CD processes are quite complex or involve large volumes of deployment.

In conclusion, by comparing the two platforms, it is possible to note that Azure has a more conservative and reliable pricing approach, which can be suitable for organizations in terms of stable and clear cost analysis. However, a pay-as-you-go model provides the advantages that AWS uses, and it is most probably suitable for small and unpredictable cases. Instead, the pricing model must be tailored depending on the organizations' needs, scale, and usage, and what suits well their financial and operational model.

Table 4: Monthly Cost Comparison

Service	Azure Pipelines (Monthly)	AWS CodePipeline (Monthly)
Basic Pipeline	$40	$20
Advanced Features	$70	$60
Total Cost	$110	$80

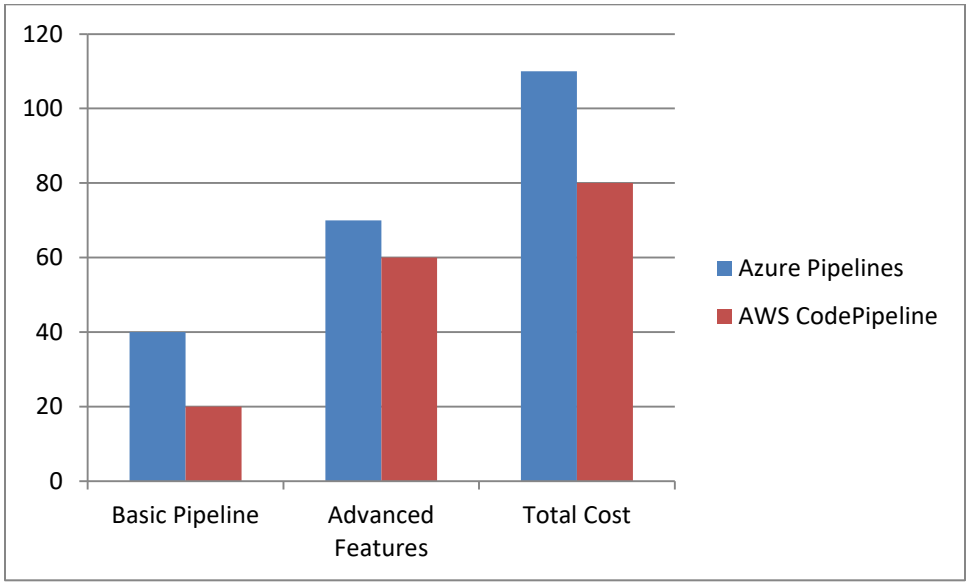


Figure 6: Monthly Cost Comparison

D. Flexibility and Performance

AWS CodePipeline has a relatively excellent reputation in terms of the level of flexibility it offers in deployment. It works harmoniously with an extensive range of deployment settings, from serverless setup as AWS Lambda to container settings as ECS. This huge built-integration enzyme enables applications to be deployed across architectures with a considerable amount of flexibility. Serverless and containerized applications are smoothly integrated with AWS CodePipeline, which helps to make fast provisioning and deployment of program workloads, especially for dynamic and flexible environments. This flexibility leads to a shorter cycle of deployment and optimum management of complicated and dispersed systems.

In turn, Azure Pipelines are much more customizable, but they can be more oriented toward the traditional deployment environments, such as virtual machines. Even though Azure Pipelines have a high level of performance and flexible settings for different deployment topologies, there is not as seamless integration with serverless and containerized systems as in AWS CodePipeline. This focus can produce a slight disadvantage of longer provisioning times for applications that are built with the

new age cloud-native architectures. Still, Azure Pipelines is useful for enterprises that primarily work with virtual machines or require a very custom deployment approach. Still, regarding deployment options, Azure Pipelines does not offer as much variety as AWS, but it represents a reliable, stable tool in the sphere it is designed for.

Table 5: Deployment Flexibility Comparison

Deployment Environment	Azure Pipelines	AWS CodePipeline
Serverless	Limited	Extensive (AWS Lambda)
Containers	Basic	Extensive (ECS)
Virtual Machines	Strong	Moderate

V. CONCLUSION

As mentioned in this paper, in order to create payment gateways, CI/CD automation from Azure enhances options through AWS. Two site design solutions are yet rather adequate and efficient, and they meet the requirements of various clients and variants. AWS CodePipeline is most popular for security, flexibility, and serverless technology compatibility, especially with AWS Lambda. For this reason, such characteristics make AWS CodePipeline ideal for automating payment gateway deployment, especially for organizations that worry much about embracing security and scalability. On the other hand, Azure Pipelines is more compatible with Microsoft products and services and has a more manageable pricing model. This makes Azure an even doubly beneficial option for companies that are already running on Microsoft services and systems – the workflows are then seamless, and the cost supported is only minimal. Finally, it is crucial to mention that the decision to choose the Azure or AWS platform for CI/CD automation should fit organizational requirements such as security requirements, budget, and tool compatibility.

A. Future Work

Subsequent studies can focus on the interconnections of Azure and AWS for leveraging their benefits in the multi-cloud conceptions. This is because integrating services from both platforms will enable organizations to gain flexibility, avoid getting locked into a specific provider, and leverage additional superior features that the two cloud providers have to offer. Such studies could include efforts to determine how to effectively manage a multicloud implementation, acquire insights into how to improve performance across the multiple platforms involved or find a means of integrating services from multiple cloud interfaces. Furthermore, one might consider the obtention of a more detailed cost breakdown to be advantageous, particularly in the case of giant businesses that are already beginning to function at scale. This research would seek to shed much more light on the long-run cost comparison between the two clouds in relation to price per volume, price for enterprises and total price tag attached to Azure and AWS, respectively. Such a direction of future work may provide useful recommendations in terms of further CI/CD pipeline optimization for organizations while considering costs.

VI. REFERENCES

[1] Saarenpää, J. (2020). Creating an Azure CI/CD pipeline for a React web application.

[2] Stefanovic, V., & Katinski, M. (2021). Pro Azure Administration and Automation. Apress.

[3] Galiveeti, S., Tawalbeh, L. A., Tawalbeh, M., & El-Latif, A. A. A. (2021). Cybersecurity analysis: Investigating the data integrity and privacy in AWS and Azure cloud platforms. In Artificial intelligence and blockchain for future cybersecurity applications (pp. 329-360). Cham: Springer International Publishing.

[4] Kaushik, P., Rao, A. M., Singh, D. P., Vashisht, S., & Gupta, S. (2021, November). Cloud computing and comparison based on service and performance between Amazon AWS, Microsoft Azure, and Google Cloud. In 2021 International Conference on Technological Advancements and Innovations (ICTAI) (pp. 268-273). IEEE.

[5] Saraswat, M., & Tripathi, R. C. (2020, December). Cloud computing: Comparison and analysis of cloud service providers-AWs, Microsoft and Google. In 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART) (pp. 281-285). IEEE.

[6] Zampetti, F., Geremia, S., Bavota, G., & Di Penta, M. (2021, September). CI/CD pipelines evolution and restructuring: A qualitative and quantitative study. In 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 471-482). IEEE.

[7] Gupta, B., Mittal, P., & Mufti, T. (2021, March). A review on Amazon web service (AWS), Microsoft Azure & and Google Cloud Platform (GCP) services. In Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India.

[8] Kamal, M. A., Raza, H. W., Alam, M. M., & Mohd, M. (2020). Highlight the features of AWS, GCP and Microsoft Azure that have an impact when choosing a cloud service provider. Int. J. Recent Technol. Eng, 8(5), 4124-4232.

[9] Ucuz, D. (2020, June). Comparison of the IoT platform vendors, Microsoft Azure, Amazon web services, and Google Cloud, from users' perspectives. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-4). IEEE.

[10] DevOps Automation in AWS and Azure: Differences and benefits, Aspire Systems, online. https://blog.aspiresys.com/infrastructure-managed-services/devops-automation-in-aws-and-azure-differences-and-benefits/

[11] Arachchi, S. A. I. B. S., & Perera, I. (2018, May). Continuous integration and continuous delivery pipeline automation for agile software project management. In 2018 Moratuwa Engineering Research Conference (MERCon) (pp. 156-161). IEEE.