STACKSPORTS DEVOPS

ABOUT STACKSPORTS

StackSports is a leading provider of **SaaS-based software solutions**, catering to over **20,000** North American sports clubs, leagues, and governing bodies. They offer sports and league management solutions and user-friendly software products that efficiently manage and run sports activities, practices, and league events.

Over the years, they have successfully assisted numerous sports organizations in managing associations, clubs, and leagues through their registration software, saving valuable time and energy.

TABLE OF CONTENTS

PROJECT OVERVIEW

In recent years, DevOps has gained significant importance in defining the agility of organizations. It enables organizations to release products incrementally with minimal risks, optimize release management costs, and enhance team productivity. DevOps combines development and operations processes, making them more efficient and agile.

Thus, we assisted StackSports in harnessing the power of DevOps to make their application development and deployment more agile, rewarding, and productive. By implementing DevOps, we focused on increasing productivity, optimizing processes, and utilizing **Continuous Integration and Continuous Delivery (CI/CD), AWS** services, structured processes, automation, and collaboration. These efforts led to improved quality and simplified management of customer expectations.

BUSINESS REQUIREMENTS & GOALS

To remain competitive and better serve their customers, our client aimed to incorporate DevOps into their software development life cycle and enhance development and delivery speed. They aimed to introduce agility into their existing development lifecycle through DevOps culture, optimize processes, and increase productivity by shortening development cycles, improving response time, and achieving successful agile delivery. To fulfill these goals, they required:

- DevOps implementation using CI/CD.
- Adoption of Microservices.
- Agile planning for incremental releases.
- Containerization.
- Monitoring for real-time visual metrics, logs, and alerts.

THINKSYS

WHY THINKSYS

As a leader in the field of **SaaS-based software solutions**, our client sought rapid deployment of high-quality software and efficient management of customer requirements.

They needed an experienced and dedicated team with expertise in cloud-based services, including assessment, migration, consulting, managed services, and seamless DevOps implementation.

Our DevOps team precisely met these requirements, working tirelessly with the developers and operations team. Leveraging our deep understanding of DevOps, CI/CD, AWS, and various other technologies, we provided effective solutions around the clock.

OBJECTIVE

Our dedicated team, an **AWS Solutions Architect, DevOps Engineer, Team Lead, and L1 Monitoring Engineer**, worked together to establish a successful DevOps setup. They followed a series of steps to ensure its implementation:

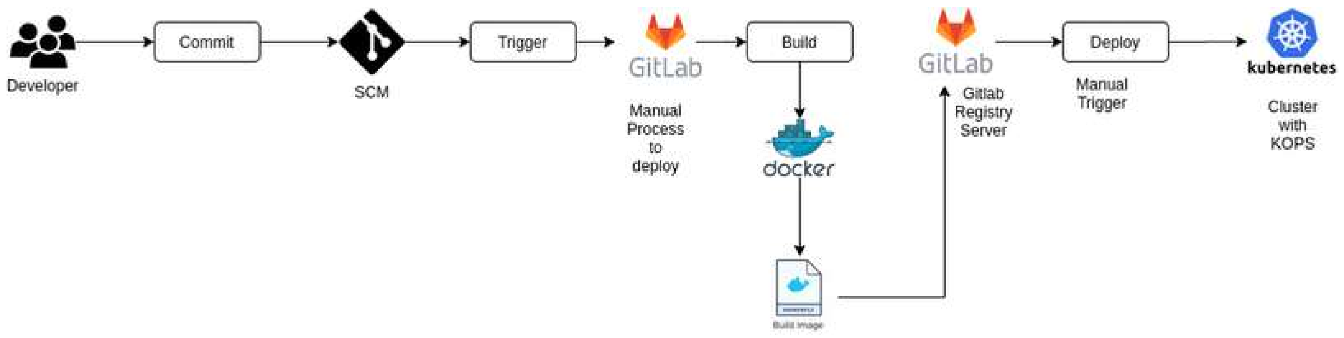
1. **Reviewing the Existing Process:** The team thoroughly examined the existing process to understand its strengths and weaknesses. This analysis provided valuable insights for improvement.
2. **Integrating Tools into the Working Environment**: The team integrated various tools into the working environment to enhance efficiency and collaboration. These tools facilitated seamless communication and streamlined workflows.
3. **Designing Procedures for System Troubleshooting and Maintenance:** Well-defined procedures were designed to handle system troubleshooting and maintenance tasks. This ensured a systematic approach to resolving issues and maintaining the system's health.
4. **Performing Root Cause Analysis for Production Errors:** The team conducted a comprehensive root cause analysis when production errors occurred. This involved identifying the underlying causes of the issues to prevent their recurrence in the future.
5. **Selecting the Correct Tool for Specific Requirements:** The team carefully evaluated different tools and technologies to select the most suitable ones for specific requirements. This approach ensured optimal tool selection based on the needs of the project.
6. **Eliminating Unnecessary Expenditure on Maintenance and Upgrades:** Efforts were made to reduce unnecessary costs associated with system maintenance and upgrades. The team implemented strategies to optimize resource allocation and eliminate unnecessary expenses.

Following these steps, our team successfully implemented a robust DevOps setup, improving efficiency, streamlined processes, and cost savings.

THINKSYS

PRECONDITIONS /TRIGGERS

Before initiating the DevOps implementation process using Continuous Integration and Continuous Delivery/Deployment, the team identified several pre-conditions and triggers that could significantly influence the overall process. Some of these factors included:



- K8s infrastructure was created by Kubernetes.
- The GitLab pipeline was used to manage deployment.
- No code tests were performed during deployments in a few projects.
- There was no vulnerability test before using the Docker image.
- No automated notification was implemented at process completion.
- NewRelic monitoring tool.

CHALLENGES

During the adoption and implementation of DevOps for the client, several challenges emerged that posed hurdles to its successful execution. The key challenges encountered were as follows:

- Migrating legacy environments to containerization.
- Creating GitLab EKS Cluster and integrating Kubernetes Cluster with GitLab Cluster.
- Setting up GitLab Runners so that our pipelines can run on it.
- Keeping connection private between Gitlab and EKS.
- Accessing the Kubernetes Cluster only through VPN.
- Selecting which projects needed dedicated runners and which could run with shared runners for cost optimization.
- Storing artifacts and files in GitLab and setting the retention policy for those.
- Setting up Cluster Autoscaler on EKS so that spot instances will come up automatically whenever there are pods that were pending due to the non-availability of resources.
- Changing the deployment methodology from blue-green to a rolling update policy to ensure zero downtime while deploying new code on the clusters.
- Setting up a readiness and liveness probe to ensure requests aren't sent to the pods that are not ready to receive traffic or are termed unhealthy.
- The technologies being used should be compatible with existing infrastructure.
- Choosing the best tools for automation and container security.
- Implementation of security on infrastructure.
- Building the infrastructure by automation and using AWS Managed services (IAAC).
- Setting up proper notification to the concerned team.
- Setting up an ALB ingress controller for exposing our application with proper security annotations.
- Achieving cost optimization by using a mixed instance policy (spot instances and on-demand instances).

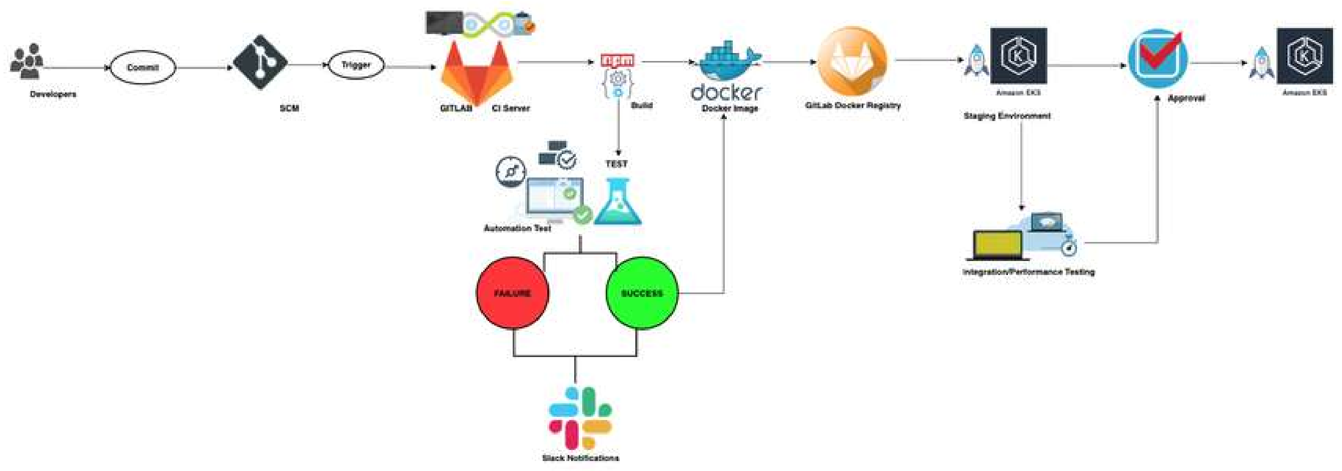
THINKSYS

PROPOSED SOLUTIONS

The DevOps team collaborated closely with the developers and operations team to comprehensively understand the existing infrastructure and code processes. This collaborative effort played a crucial role in successfully transitioning to DevOps practices.

The development and operations teams received dedicated DevOps training to facilitate a smooth transition. This training gave them a holistic understanding of the DevOps Ecosystem, including the processes, tools, and best practices. This knowledge enabled the teams to navigate their respective roles and responsibilities seamlessly.

By fostering a culture of continuous learning and knowledge sharing, the teams embraced the principles of DevOps and worked together more efficiently. The training sessions facilitated effective communication and collaboration, enabling the teams to align their workflows and streamline their processes.



Our team also worked transversally to observe, identify, and correct practices whenever needed and proposed the following solutions to overcome the challenges encountered during the project assessment and implementation phases:

- We overcame the Developer vs Operational team mentality and worked in collaboration.
- Migrated the infrastructure from on-premises to Cloud.
- Automated test cases in various deployment tools.
- Moved from legacy infrastructure to Microservices.
- Upgraded Kubernetes by using EKS AWS.
- Automated the infra-creation process using Terraform.
- Upgraded the pipelines in GitLab according to the new deployment strategy.
- Implemented auto vulnerability test utility on docker hub images before using Configure Slack notification services on code commit and at process completion in the pipeline.
- Configured Prometheus for K8s monitoring with Slack notification.
- Enhanced NewRelic monitoring by enabling distributed monitoring.
- VPC peering was used to keep our connection private between GitLab and EKS.
- Access to Kubernetes Cluster was ensured by keeping Cluster Endpoint private and attaching Security Groups to the control-plane security groups.
- Implemented a continuous improvement framework.

TECH STACK

Listed below are the various tools and technologies used by the team for automating processes at various phases of the DevOps project life cycle:

TECHNOLOGY/TOOL	PURPOSE/USE
Kubernetes	Managed containerized workload and services.
Kops	Managed Kubernetes clusters.
EKS	Used to manage and deploy clusters.
EC2	Cost-effective tool used for its compute capacity and ability to configure security and networking, and manage storage.
Load Balancer	Used to increase capacity (concurrent users) and reliability of applications.
S3	Used to store Kops state file, as it offers access to a highly scalable, reliable, fast, inexpensive data storage infrastructure.
AWS Lambda	Used to trigger functions and manage underlying compute resources.
AWS RDS	Hosted the database server.
SQS	Transmitting volume of data at any level of throughput and decoupling application components.
Cloudwatch	Infrastructure monitoring.
ELK	Used for data collection, processing, and analysis.
Opsgenie	Incident management.
GitLab	Used for Versioning Control.
GitLab CI/CD	Used for both versioning control and CI/CD.

OUTCOME

Based on the client's requirements, our team found that AWS Cloud Service was the correct choice as it was cost-effective and offered needed security. Moreover, we adapted and learned various DevOps tools to ensure smooth DevOps adoption and CI/CD implementation and took the necessary measures to ensure higher effectiveness of the software due to frequent iterative releases.

In short, after a successful implementation of DevOps culture in the software development life cycle, the client was able to enjoy a variety of benefits, such as:

1. **Easy Automation:** The introduction of DevOps practices enabled the automation of various manual processes, leading to increased efficiency and productivity.
2. **Effective Monitoring:** Robust monitoring tools were implemented, allowing for real-time visibility into the system's performance and the ability to address any issues or bottlenecks proactively.
3. **Faster Time-to-Market:** The streamlined deployment process, facilitated by CI/CD practices, significantly reduced the time required to bring new features and updates to the market, giving the client a competitive edge.
4. **Better Release Management:** DevOps practices, such as automated testing and deployment pipelines, improve the management of software releases, ensuring smoother and more reliable deployments.
5. **Reduced Production Release Time:** Through automation and continuous integration, the time taken for production releases was significantly reduced, enabling faster delivery of software updates and enhancements.
6. **Cost Reduction:** The adoption of DevOps practices led to cost savings of up to 30% while providing an efficient CI solution for the software development process.
7. **Accelerated Service Implementation:** The implementation time for new services was reduced from months to minutes, allowing the client to adapt to changing market demands quickly.
8. **Optimized Cloud Infrastructure:** By leveraging AWS Cloud Service, the client ensured that their investments in cloud infrastructure, analytics, and data management were utilized effectively, avoiding wasteful spending.

Overall, the successful implementation of DevOps practices brought numerous benefits, empowering the client to achieve greater efficiency, agility, and cost-effectiveness in their software development and deployment processes.

THINKSYS

LET'S WORK TOGETHER

info@thinksys.com

www.thinksys.com

440 N Wolfe Road, Suite #22
Sunnyvale, CA 94085