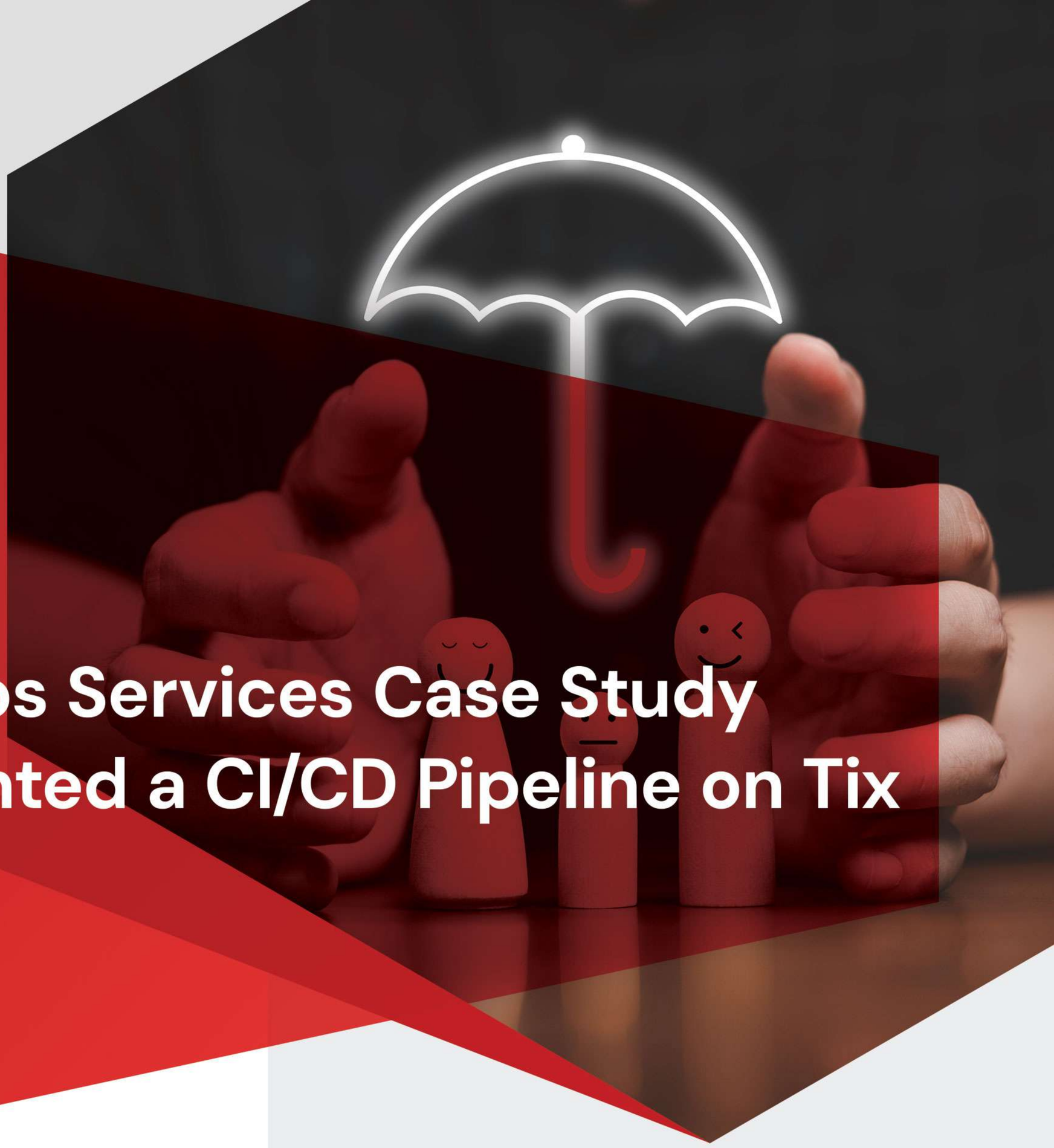
rapyder
Solutions, Simplified.



AWS DevOps Services Case Study – Implemented a CI/CD Pipeline on Tix

Introduction

Tix is a ticket booking application based in Australia with a single goal of making live events more affordable & accessible. They take out the pain of searching different sources to find tickets and put them all in the palm of your hand with the TIX app. The application operates in 6 cities across 5 countries with over 1.60 million members & app users.

Business Need

In this competitive era, every business aims to provide its customers with the best product and service in no time. TIX aims to create one platform where live entertainment can be affordable and more accessible. For this, the client was developing a B2C online booking application to be hosted on AWS cloud and had two prime requirements:

- They were looking for AWS expertise that would provide solutions on AWS and help them implement the entire Continuous Integration and Continuous Deployment (CI/CD) with no manual intervention and auto rollbacks in the case of deployment failures with AWS services
- A highly scalable and reliable architecture for their application.
- A CapEx-based solution instead of an OpEx for the application to change the application's architecture later if needed.

Solution Approach

- They were looking for AWS experts that would help them provide solutions for shortening their development cycle with reduced deployments and auto rollbacks.

- The client had 3 different environments, Dev, SIT, and Production Environment. For each pipeline, unique AWS CodePipelines were implemented for automated deployments.

- WebHooks were configured for each pipeline so that the pipelines would get triggered with every commit.

- AWS Parameter Store was used to store & secure the configuration data, which will be replaced at the time of deployment based on the environment.

- Jenkins was configured for MS builds which will push the binaries to AWS S3 to manage, organize and secure the data.

- A package manager server was also set up from where the application can download the Node and NuGet packages, not making it available to the public internet for downloading the packages. This, in return, reduced the build time.

- Code Deploy was used to deploy the artifact to the target servers to avoid downtime during the deployment.

- Notifications were set up at the pipeline level to notify clients about new deployments and failures.

- Rollbacks were configured; the previous stable version will be deployed in case of a failed deployment.

- CloudFormation scripts were written for infrastructure to replicate the architecture in a short period.

Reaping Rewards

The entire process for all the 3 environments was successfully implemented in the time frame of 1.5 months with full automation. After implementing the solutions provided by Rapyder, TIX witnessed sustainable results for their application development.

Here's a low down of benefits derived:

- With the help of CloudFormation, the developers could replicate the whole architecture and launch it in no time.

- A fully automated CI/CD solution eradicated the risk of manual-prone error entirely and fastened the development cycle.

- 30% of developers' time is now invested in coding instead of manual deployments and resolving bugs.

- Manual implementation for a CI/CD solution would have taken around an hour, and de-bugging an error would take the whole day. But with a fully automated CI/CD solution, the process was completed in 10 minutes.

- The CloudFormation script (Infrastructure as Code) made the development and deployment process efficient, enabling faster go-to-market

rapyder
Solutions, Simplified.

+91 733 868 6644
info@rapyder.com
www.rapyder.com