Adopting the CI/CD Methodology

Presented by Rob Maracle

Overview

- Understanding the challenge
- Introducing CI/CD
- Culture shift
- Profit

Understanding the Challenge

- Many steps required to ensure code quality, all of which tie up developer's time. (e.g Static code analysis, Compile Time, Vulnerability scanning, etc.)
- Administrative overhead for copying files, testing deployment, and deploying to production
- All these steps are prone to human error, and can only be done as fast as a human can perform them.
- The time taken by developers and admins to perform these takes results in unpredictable payroll spend, and the cost of compute/storage resources used while admins and devs are using them to QA, and deploy their code (sitting idle over nights and weekends, for example).

Introducing CI/CD

- Continuous Integration, and Continuous Deployment is an Automated, Version Controlled, Repeatable process that frees up developers and administrators to get back to their core tasks.
- By automating the code quality assurance and deployment routines, we save payroll time, and the compute resources used for these purposes are only needed as long as the automated job needs them for, saving money on infrastructure too!
- By having a script run these tasks, they become perfectly repeatable, removing the element of human error, which further reduces payroll spend, and time to market!

In Conclusion

- Everybody wins with CI/CD!
- The developers spend their time writing new code.
- The admins spend their time focusing on real production tasks.
- Infrastructure costs are reduced.
- The code quality is improved.
- New releases of code become trivial, and can happen at any time, with little effort.