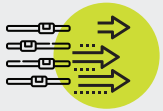
DevOps Case Study

BACKGROUND

This case study aptly denotes the improvement in the release management and deployment activities for the IT Dept, in an large retail organization, based out of Europe.

The IT unit had developed various applications, but due to the complex landscape of IT infrastructure, the release involved build and deployment of multiple applications and its decencies.

Release management challenges



Non-uniform processes

- Various development teams had used disparate tools and modes for build and deployment.
- While some teams build project artifacts (.war, .ear files) from IDE's, some used ANT scripts , where as few others used Maven scripts.
- Due to this variance the release and deployment process varied across projects.

Manual activities

- Activities such as
  - Uploading files
  - Release labeling, and
  - Code packaging were done manually.
- Deployment, too was done manually using a release document.
- Due to the above issues handled manually, each release encountered regression during deployment.

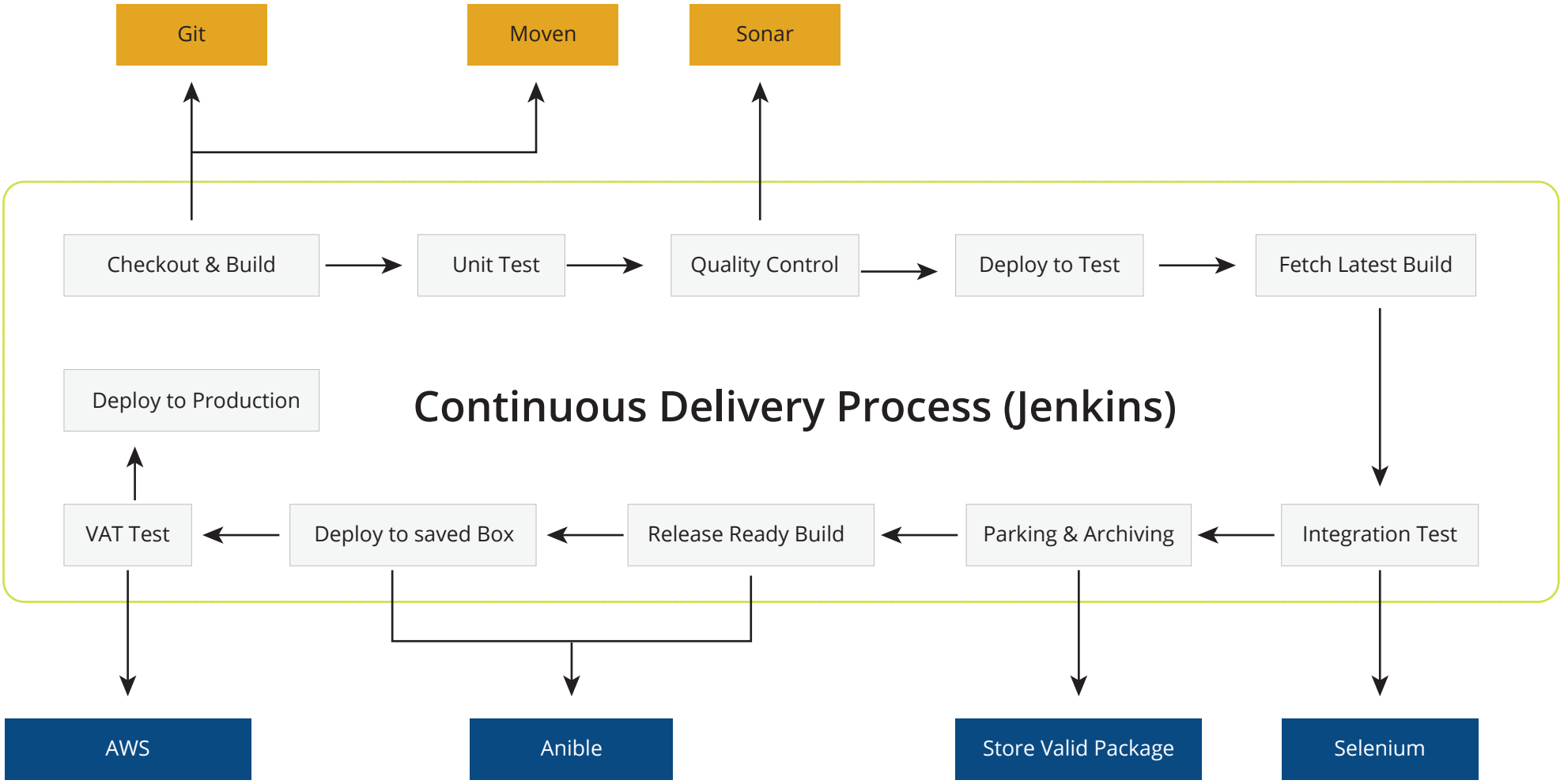
Absence of automated validation

- Since most of the testing activities were done manually, it resulted into – impacting the release timelines.

Lengthy release times

- Due to these above factors, production releases even for a small enhancement, would take almost 6 hours!

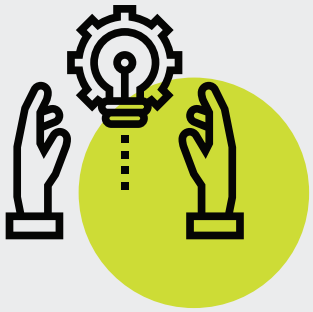
Implementation Diagram



DevOps comes to rescue

In order to address these issues, DevOps processes were set up, enabling a smoother touch to the processes as given below:

- Build and deployment processes across all the projects, were made consistent through Maven scripts, which also allowed the reuse of existing scripts.
- Jenkins CI was used as the continuous integration tool to develop a robust deployment framework. This CI tool reduced many of the manual activities such as manual build, manual testing, manual packaging etc.
- Build and testing was automated using Jenkins plugins. Junit and selenium frameworks were used to automate unit and web testing.
- Jenkins dashboard was used as a unified project dashboard to monitor project status, build failures, code coverage etc.
- Micro services implementation for 4 of their core products.
- Notification plugin was used to alert the administrator in case of build failures.



Major Outcomes

- Automation and continuous integration reduced the average production release time to 30 mins to 20mins and now to 10-11mins from 6 hrs.
- This started 1 product – now we are running the same best practices for more than 4 products which are live , which has resulted in enormous savings in efforts.
- Received President's Award for implementing efficient and effective CI/CD pipeline.