



Mindtree

A Larsen & Toubro Group Company



# Building an In-house DevOps Platform

*DevOps is an integral part of all modern applications. DevOps automates the release management activities along with quality control and productivity improvement measures. In this white paper, we discuss building an DevOps platform for a mobility solution.*



## Introduction

DevOps promotes a culture of collaboration, fosters the philosophy of iterative releases, and supports the agile methodology. DevOps practice complements the agile methodology through improved and automated quality measures. We could also explore other automation possibilities such as automated testing and automated release management in the DevOps practice.

DevOps is a culture, movement, or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.

### DevOps Advantages



Increased release velocity due to continuous development and integration methodology

Increased productivity due to automation

Increased delivery quality due to automated quality controls

Reduced risk due to automated quality controls

Improved performance due to automated monitoring



## Key tenets of DevOps

Given below are the main tenets of DevOps practice:

- **Continuous Build:** In this practice, the developers continuously develop the code and check into a centralized repository for building the feature or a story. We setup the governance for automated quality checks.
- **Continuous Integration:** This is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.
- **Continuous Deployment:** In continuous deployment, every change that passes the automated tests is deployed to various environments automatically.
- **Continuous testing:** This is a practice where the new version of an application is rigorously tested to ensure that it meets all desired system qualities. It is important that all relevant aspects - functionality, security, performance, or compliance are verified. We continuously run automated unit tests, functional tests, and regression tests.
- **Continuous Monitoring:** This practice helps identify issues or bottlenecks in the end-to-end pipeline and helps make the pipeline effective. We carry out various automated monitoring practices such as server health check monitoring, application monitoring, performance monitoring, security monitoring, and such.
- **Iterative release:** As part of the agile delivery model, we deliver the features in various iterations
- **Modular development:** The code components are developed as loosely coupled, reusable, independent modules.
- **Automation:** We need to automate the key activities such as code review, build, deployment, testing, and monitoring.

## Need for In-house DevOps

Streamlined and automated CI/CD bringing in no licensing cost by utilizing inhouse / open source tool, provides a readymade framework for any new project to bootstrap quickly, which will largely influence the ability to meet timeline with quality delivery. We have depicted a sample of continuous integration (CI) and continuous deployment (CD) processes for in-house DevOps in Figure 1.

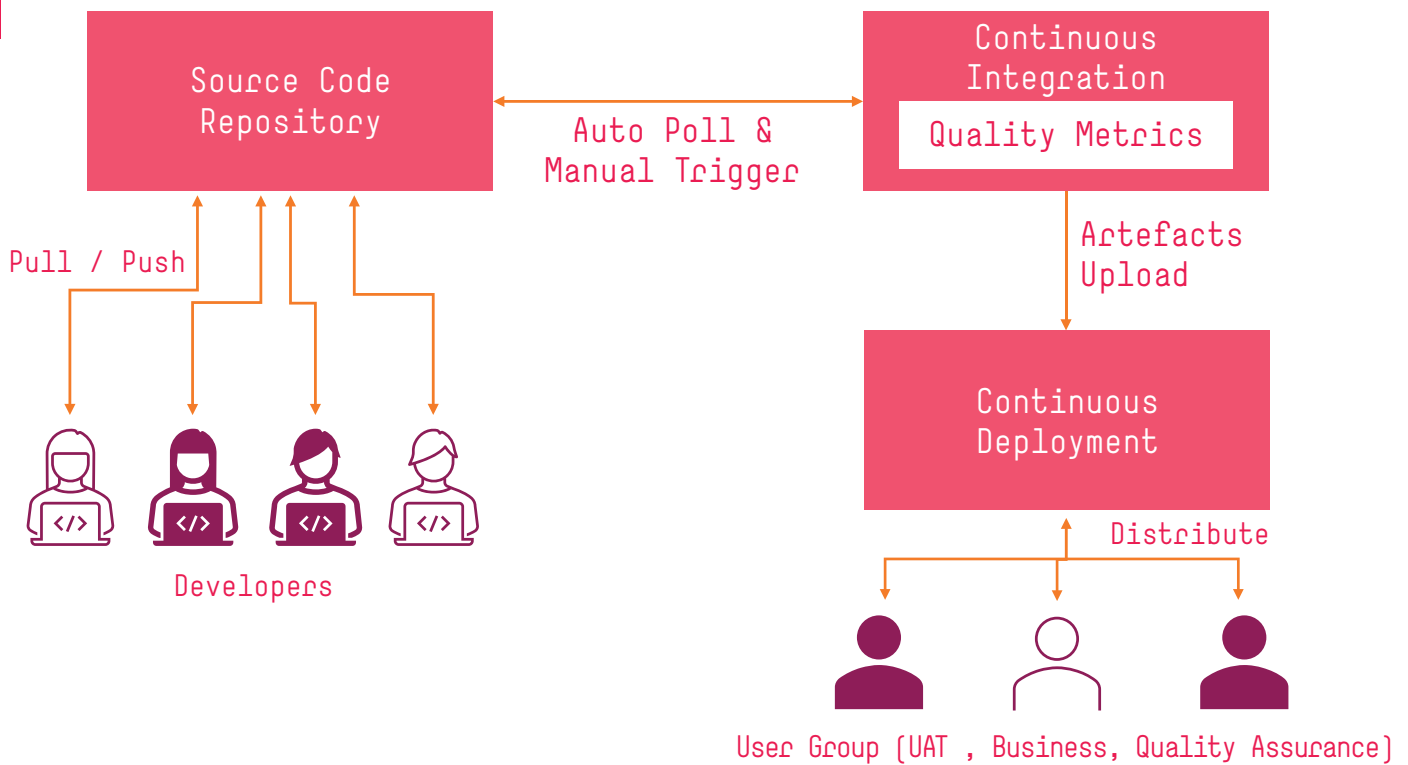


Figure 1: InHouse DevOps overview

## In-house DevOps

In this section, we shall look at the main design considerations, drivers and motivations for In-house DevOps model.

In-house DevOps essentially involves building a DevOps platform within an organization using open source DevOps tools. This provides us with greater control over the DevOps pipeline and processes. In few scenarios, due to data security and privacy concerns, organizations restrict the data that is sent out of the organization, in such cases In-house DevOps comes in handy.

### Drivers and Motivations

Essentially In-house DevOps involves leveraging the organizational knowledge and open source tools to build a robust DevOps platform. Let us look at various aspects of the In-house DevOps.



## Advantages

The main advantages of In-house DevOps are as follows

- In-house DevOps is easy to setup and configure
- Low cost infrastructure dependency
- Optimized license cost due to usage of open source tools
- Ability to build multiple technology based apps and services
- Flexibility to integrate, quality profiles, and multiple build flavors
- Leverage existing collaboration and communication model within the organization
- Leverage the deep knowledge of organizational policies, software to build a robust DevOps
- Quick bootstrapping projects for CI/CD
- Automated, Streamlined and quality build release management
- Greatly improves Release management with streamlined process
- Greatly improves the QA Strategy and plan with a definite timeline for build releases
- Greatly improves development time and saves time for build release activities
- Removes dependency of local PC based / developer based build release activities
- Flexibility to create new build flavor and distribution groups
- Helps to maintain the backup of build artifacts
- Helps to identify the code quality and build status for every commit sent by the development team.

### Drivers for In-house DevOps



Given below are the main drivers for in-house DevOps:

Need for open  
source tools

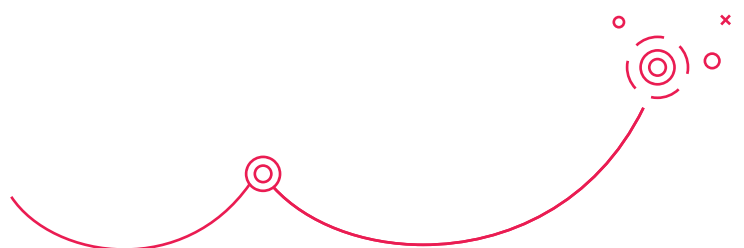
High license cost  
for available  
SaaS solutions

Make effective use  
of whitelisted  
softwares within  
organization

### Implementation



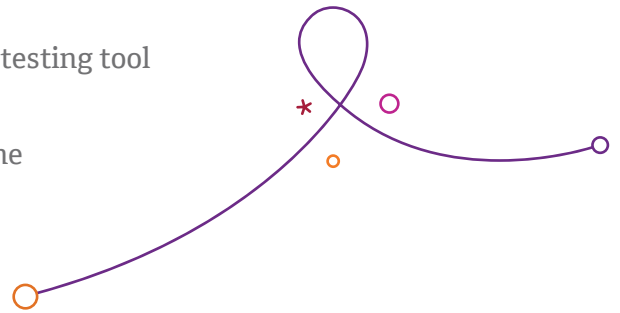
In this section, we detail the implementation details of building a robust in-house DevOps platform



# Pre-requisites for overall DevOps and Release management

Given below are the main pre-requisites for the in-house DevOps:

- Source code management system (SCM) to manage the source control
- Build server to setup the DevOps infrastructure
- Opensource DevOps setup that enables multiple technology profiles with the ability to integrate build tools, quality profiles and options to configure multiple build pipelines
- Ability to remote trigger build is a good to have option when a manual build trigger is required
- Build environment strategy including the number of build environments to consider
- Branching strategy and code review strategy
- Build steps and activities to be performed
- Build numbering strategy, source code Tagging
- Dependency manager and Build tools
- Quality profile tool and quality gate profile for the builds
- Security tool for security testing
- Testing tools such as unit testing tool and functional testing tool
- Build status alerts and notification strategy
- App distribution strategy for each build flavor/pipeline
- App promotion strategy



# Tools and strategy consideration

Given below are the various sample tools that can be used for implementing an in-house DevOps strategy:

Category	Sample Tool
Source Code Management	SVN, Gitlab, Bitbucket
DevOps	Jenkins
Build Machine	iMac
Build Technology	Android and iOS
Quality Profile	SonarLint , SwiftLink, Android Kotlin, SonarQube, OWASP Dependency Check
Build Tools	Gradle , XCode
Monitoring	Nagios, Splunk
Code Analysis	SonarQube, FindBugs, Checkstyle, JSLint
Code Coverage	Clover
CI Tools	Jenkins
Version Increment	Auto increment on patch version Manual configuration for major.minor
Alerts and Notification	On build failure with changelist On successful upload of build artefacts
Security Tool	Burp Suite, AppScan
Testing Tools	Junit, Selenium, JMeter
App Distribution	AppCenter, Playstore alpha/beta, TestFlight
App Promotion	Base on QA sign off

# Steps to configure the DevOps setup

In this section we have detailed the sample steps to configure an in-house DevOps system. The setup is mainly configured for mobility solutions.

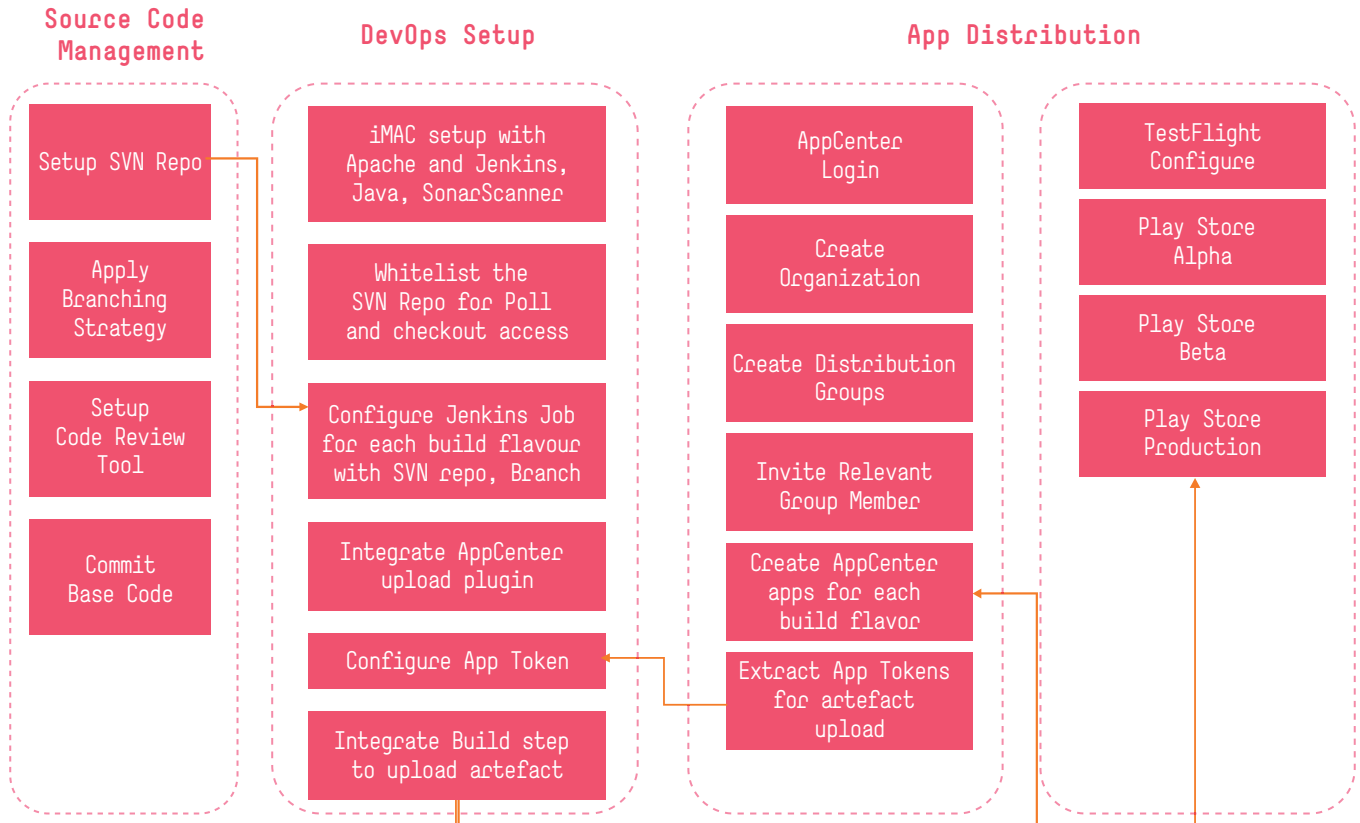


Figure 2: Steps to configure DevOps



## Source Code Management Phase

As a first step, we need to setup the source control repository such as SVN. As part of this we need to implement the branching strategy to manage the code needed across various environments and to implement the code promotion. We should also finalize the code review tool and commit the baseline code.



## DevOps Setup Phase

The second step mainly involves setting up the DevOps tools on the server. In this example we setup the Jenkins and SonarScanner. We need to setup the connection between SVN repository and the DevOps server. We can then setup the build process using Jenkins for specified SVN branches. We can install the AppCenter upload plugin to push the built artifact to the AppCenter. The extracted app tokens should be configured for artifact upload.



## Mobile app distribution phase

To distribute the app we should login to AppCenter and create the organization and distribution groups. We should invite the relevant group members and configure the AppCenter apps for each build flavor. We should configure the TestFlight, Play store Alpha, Play Store Beta and Playstore Production for app distribution.

# Best Practices

Given below are some of the key best practices of In-house DevOps model:

1. Instead of triggering build for every commit, trigger build once in every hour such that commits in last one hour get consolidated
2. Stagger the auto build pipeline trigger for any build variant by an hour apart to avoid overloading builds or making them wait in build pipeline
3. Enable remote trigger capability for flexible build trigger
4. Use environment variable, keystore for access keys, certificates, and URLs to avoid direct access in the script
5. Do not use incremental builds for any builds going to QA and other testing purpose
6. Enable artifact upload capability directly to Playstore alpha/beta or TestFlight from Pipeline

We have depicted a sample setup of In-house DevOps in Figure 3

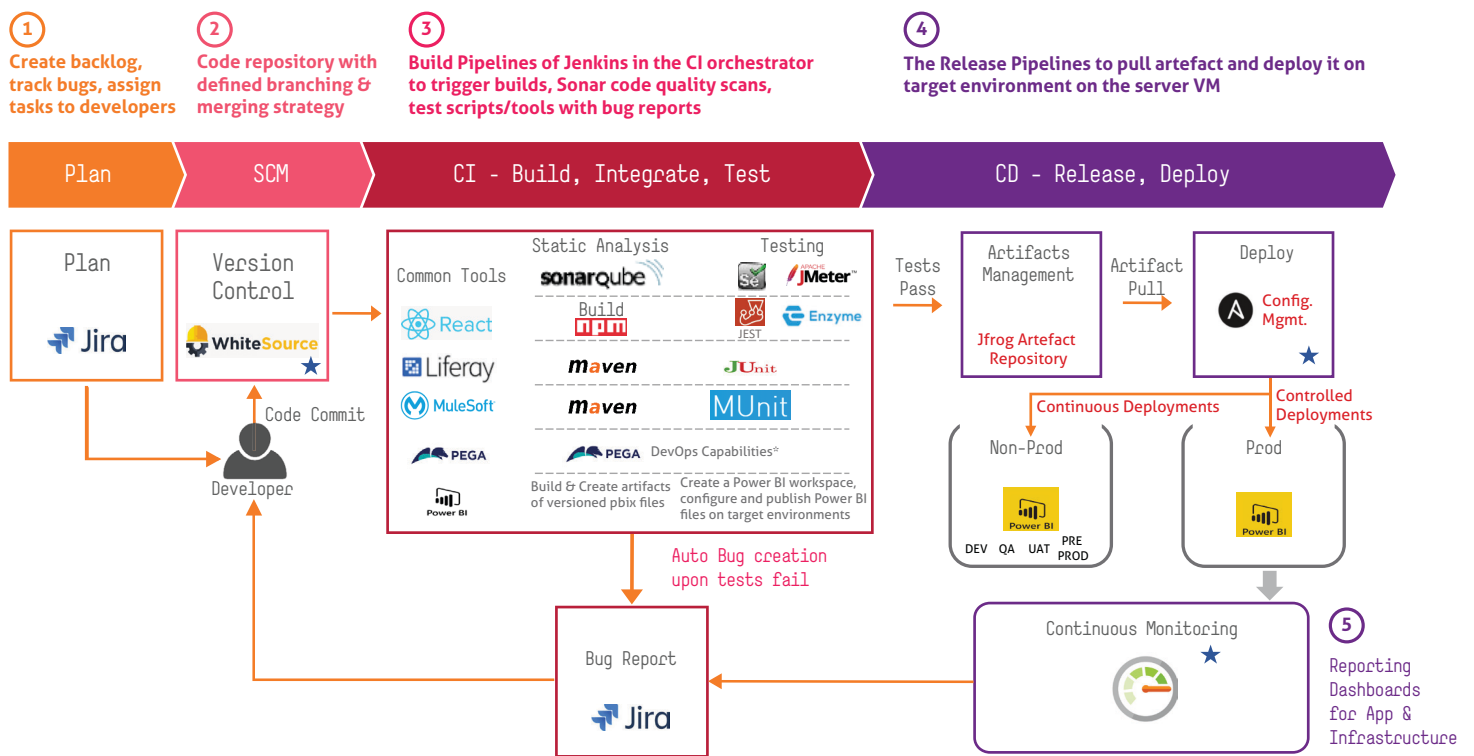


Figure 3 Sample In-house DevOps Setup



## Sandhya

Senior Architect

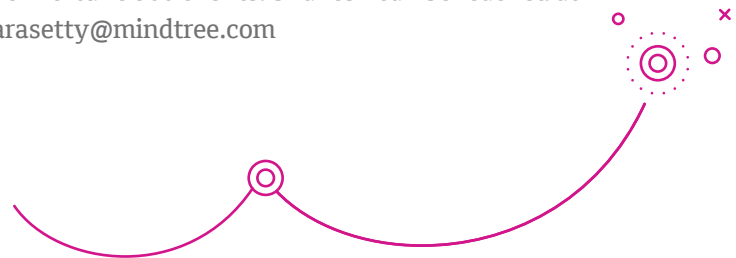
Sandhya is a Mobile Architect who has worked on multiple mobile projects with the in-house devops setup and configuration of these systems. With a hands-on experience of using in-house setup and identifying real benefits of streamlined release management, she has greatly helped in delivering quality apps to the customers.



## Dr. Shailesh Kumar Shivakumar

Solution Architect

Dr. Shailesh Kumar Shivakumar has 19+ years of experience in a wide spectrum of digital technologies including, enterprise portals, content management systems, lean portals, and microservices. Dr. Shailesh holds a PhD degree in computer science and has authored eight technical books published by the world's top academic publishers such as Elsevier Science, Taylor and Franscis, Wiley/IEEE Press, and Apress. Dr. Shailesh has authored more than 14 technical white papers, five blogs, twelve textbook chapters for various under-graduate and post graduate programs and has contributed multiple articles. He has published 20+ research papers in reputed international journals. Dr. Shailesh holds two granted US patents, apart from ten patent applications. Dr. Shailesh has presented multiple research papers at international conferences. Dr. Shailesh's Google Knowledge Graph can be accessed at <https://g.co/kgs/4YoaiN> . He has successfully led several large scale digital engagements for Fortune 500 clients. Shailesh can be reached at [Shaileshkumar.Shivakumarasetty@mindtree.com](mailto:Shaileshkumar.Shivakumarasetty@mindtree.com)



## About Mindtree

Mindtree [NSE: MINDTREE] is a global technology consulting and services company, helping enterprises marry scale with agility to achieve competitive advantage. "Born digital," in 1999 and now a Larsen & Toubro Group Company, Mindtree applies its deep domain knowledge to 260 enterprise client engagements to break down silos, make sense of digital complexity and bring new initiatives to market faster. We enable IT to move at the speed of business, leveraging emerging technologies and the efficiencies of Continuous Delivery to spur business innovation. Operating in 24 countries across the world, we're consistently regarded as one of the best places to work, embodied every day by our winning culture made up of over 27,000 entrepreneurial, collaborative and dedicated "Mindtree Minds."