MARCH 30, 2022

brillio one.ai HPE

Case Studies

LEADING AMERICAN TELECOMMUNICATION COMPANY

Templatize DevOps platform to scale

Problem statement

To help in integration of DevOps tools for automating the on-premise and cloud. Also, setting up of continuous delivery model with minimal downtime

Challenges faced

Low tool awareness

Siloed development of various attacks

Slower production deployment

Inefficient automation practices

Overcoming challenges

Metrics based quality measurement

Code coverage improved from ***30% to 85%***

Streamlined process leveraging existing tool

Centralized code and deployment scripts

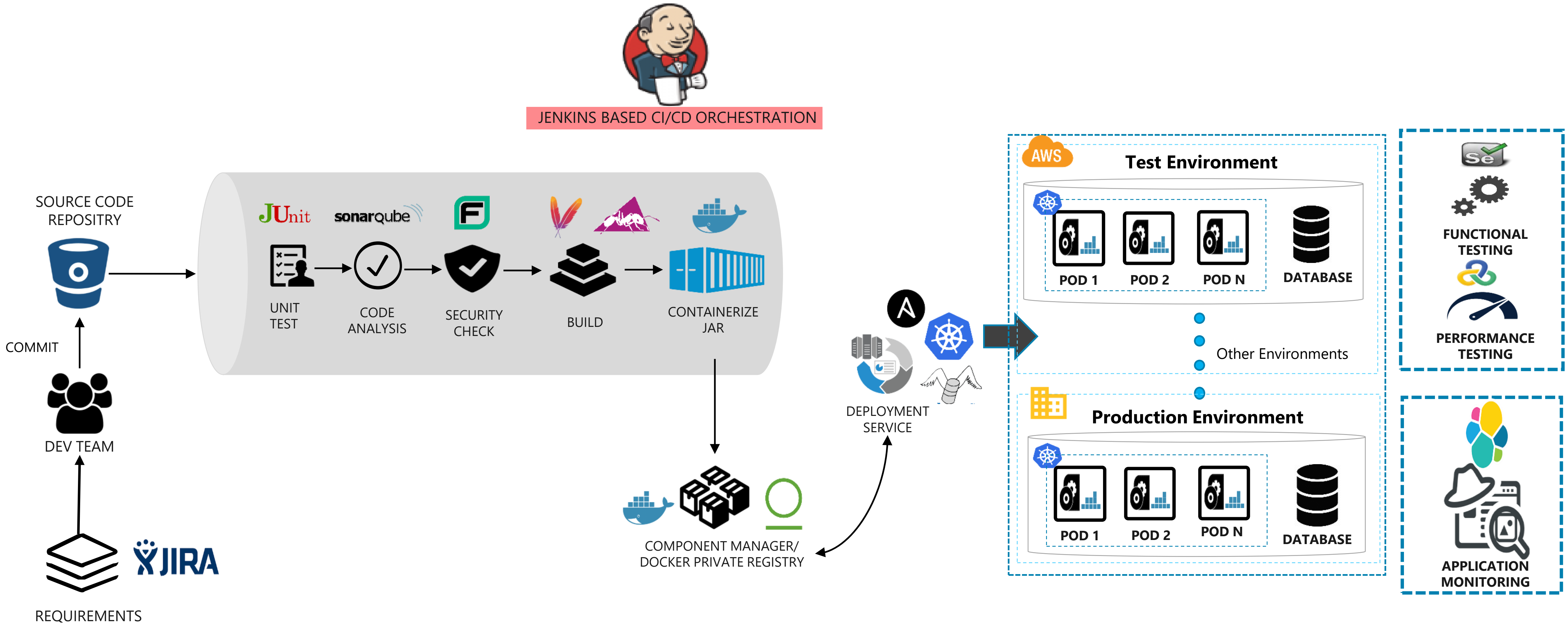
Deployment window drastically reduced from ***6 hours*** to ***1 hour***

Tech Stack

FORTIFY, sonarqube, Bitbucket, JIRA, JUnit 5, Confluence, ANSIBLE, Selenium, amazon web services, docker, kubernetes, Jenkins, ELK Stack

brillio one.ai

Our Implementation Approach



LEADING AMERICAN INDEPENDENT BANK

Unified DevOps platform for Heterogeneous multi vendor setup

Problem statement

To help transition the enterprise from in house tools to opensource technology and implementing automation tools as part of DevOps strategy

Challenges faced

Overcome the Opensource tool adoption

Team training on branching strategies

Defining common DevOps process

Automation for one integrated application

Multiple vendors across multiple technologies

Overcoming challenges

Automation executed for Mobility, MuleSoft, Salesforce, MS Dynamics

Unified Branching strategy for smoother transition to other environments

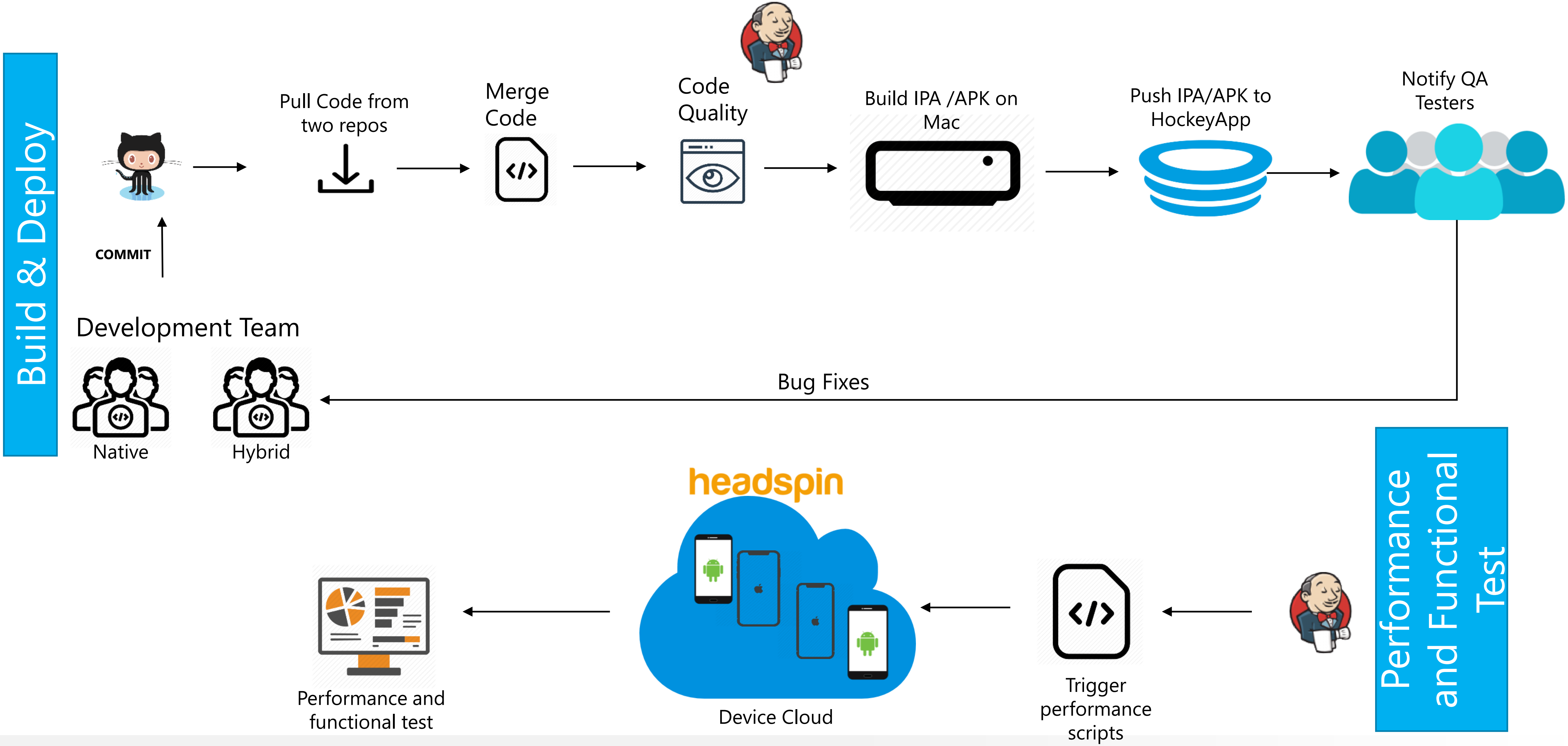
Multi step manual build process to *single step automated process*

Unit test and Functional test automation integrated into the build and deployment process

Tech Stack

Jenkins · MuleSoft · HockeyApp · GitHub · python™ · JFrog · JIRA · Confluence · appium · APACHE JMeter™

Our Implementation Approach



LEADING AMERICAN AUTOMOBILE MANUFACTURER

DevOps enabled product development

Problem statement

To help in creating multi-channel customer experience with continuous delivery framework for iOS supported devices

Challenges faced

Multiple build process for various channel

No branching strategies defined resulted in manual integration

Overhead effort in testing the mobile application

Overcoming challenges

Single click application deployment on test devices

Test overhead effort reduced from *hours to minutes*

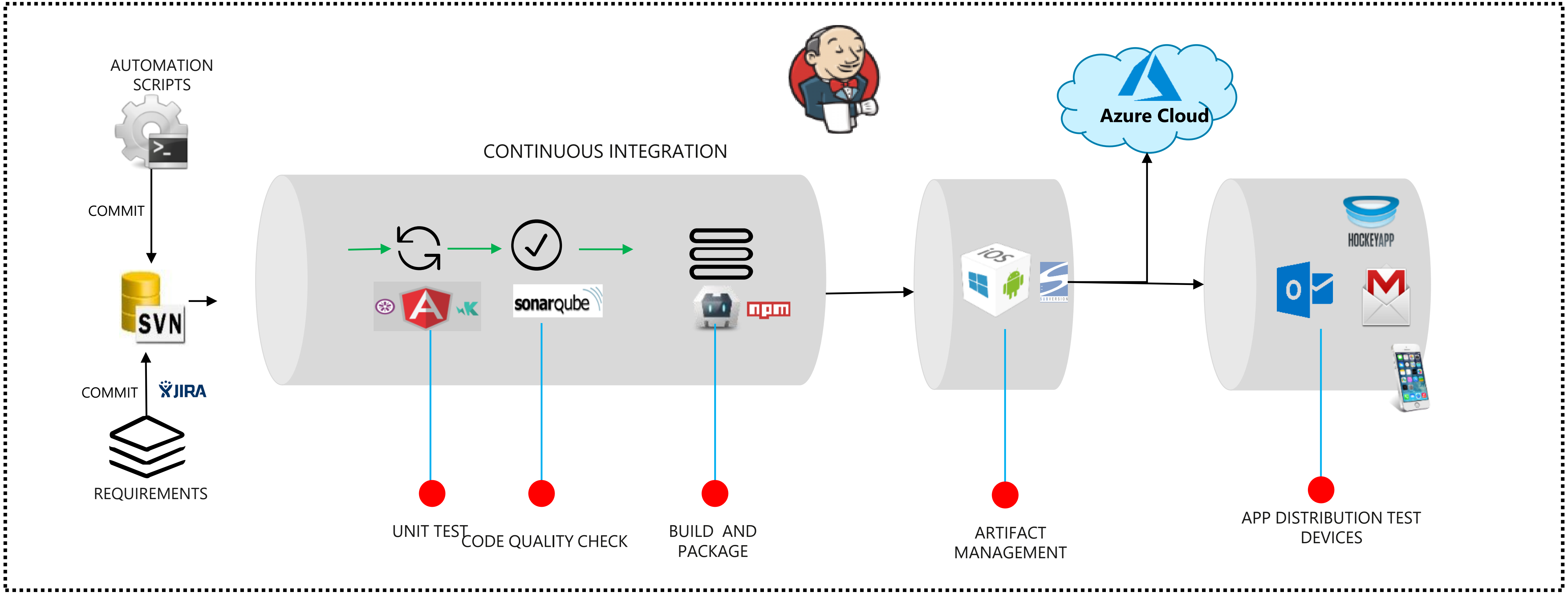
Integrated unit test automation and quality checks

Branching strategy defined to automate merge process

Tech Stack

SUBVERSION sonarqube JIRA Confluence Jasmine Jenkins HockeyApp

Our Implementation Approach



LEADING AMERICAN COLD STORAGE LOGISTICS COMPANY

Unified DevOps platform for Heterogeneous multi vendor setup

Problem statement

Implement DevOps Pipeline along with integration of Multiple processes, methods and technologies powered by intelligent insights.

Challenges faced

Overcome the tool adoption

Team training on branching strategies and DevOps Tools

Defining common DevOps process

Automation for one integrated application

Multiple vendors across multiple technologies

Overcoming challenges

Automation executed for MuleSoft, NodeJS

Unified Branching strategy for smoother transition to other environments

Multi step manual build process automated to *bot message in slack*

Unit test and Functional test automation integrated into the build and deployment process

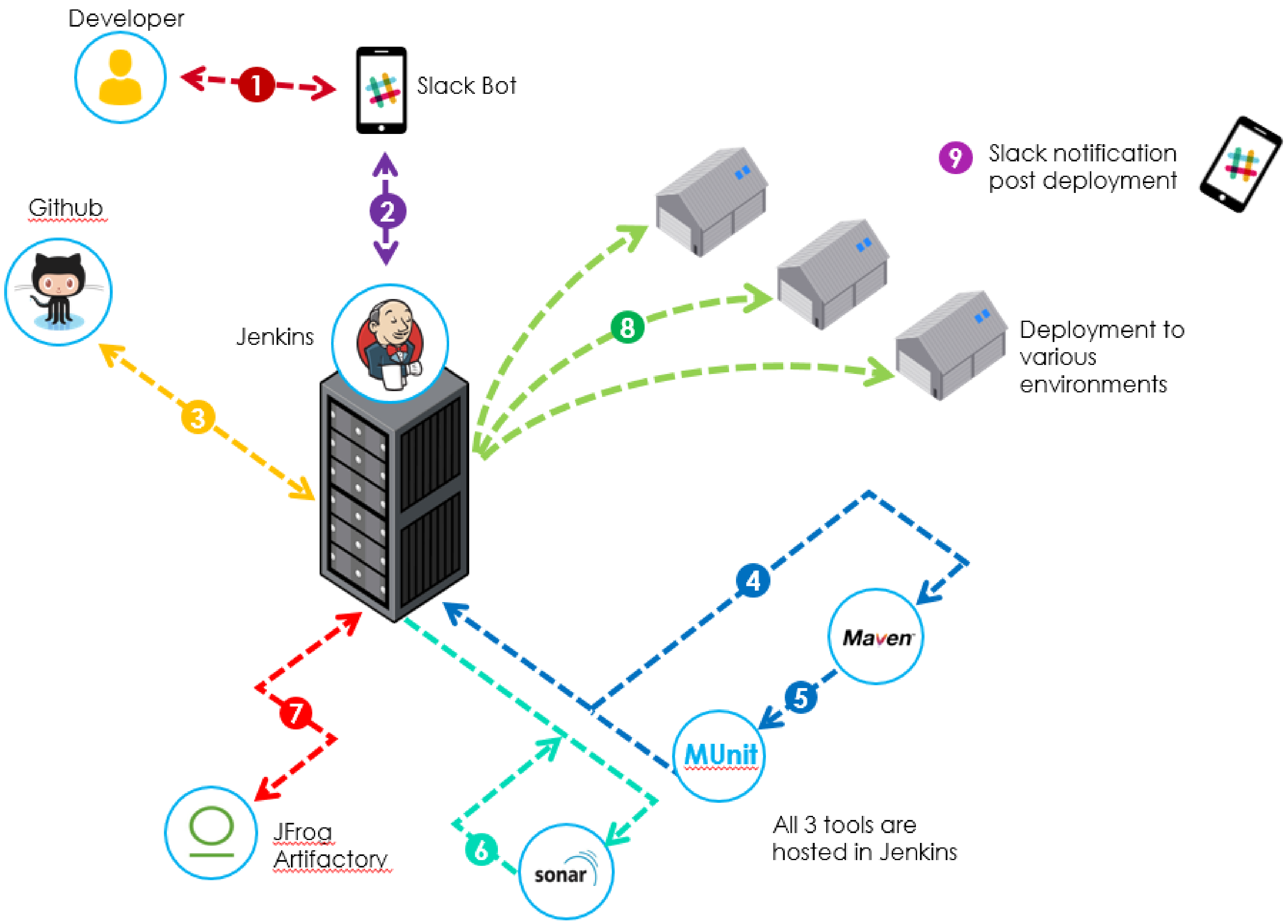
Monitor and track performance of DevOps activities and AWS Services

Tech Stack

Jenkins, MuleSoft, GitHub, JFrog, JIRA, Confluence, Apache JMeter™, QUAY, sonarqube, docker

HYGIEIA, Se, slack, solarwinds LOGGLY

Our Implementation Approach


Developer
1
Slack Bot
2
9
Slack notification post deployment
Github
Jenkins
3
8
Deployment to various environments
4
Maven
5
MUnit
7
JFrog Artifactory
6
sonar
All 3 tools are hosted in Jenkins

1. Developers message in slack to build/deploy mule Module.
2. Slack triggers respective job in Jenkins.
3. Jenkins checkout code from Github.
4. Build the code using Maven build tool in Jenkins.
5. Run tests using Munit in Jenkins.
6. Run Sonarqube code analysis in Jenkins.
7. Upload artifacts to Jfrog artifactory repository from Jenkins.
8. Deploy the zip file(created after maven build i.e. Step2) to Cloudhub from Jenkins.
9. Notification sent to Slack about the build/test/deployment details.

brillio one.ai

Thank You!

LET US CRAFT SOMETHING

AMAZING. **TOGETHER!**