

# DevOps Platform

## DevOps Platform Phase 1 Executive Overview

DevOps platform for phase 1 will focus on delivering the foundational services and minimal feature set related to Integrated Tool Chain Framework and Tool Chain Analytics. The FeatureSets are categorized into 4 sections – each being developed in parallel for the initial product version to be ready within 4 – 5 months. The business usecase delivered would allow for clients / consumers to adopt the Integrated Framework for free while there would be a subscription model for the Analytics. Below is a summary of the 2 Epics to be delivered.

**· Integrated Framework Bundle – open source UI template manager to construct a containerized tool chain on cloud/private cloud with built-in accelerators**

**· Analytics – Plugin-collectors for selected DevOps tool chain to collect activity elements and store into ELK / Pormetheus to be rendered to UI dashboard**

We would recommend going with the greenfield approach – having the integrated framework bundle deployed allowing for collection of data in a structured manner (does not require data pre-processing / cleanup) to be displayed in web-based portal. We can provide the option to adopt either product feature (IFB, Analytics) – but the analytics collectors would require us to gather more data points to construct intelligence in analyzing the dataset and identify meaningful information.

## Feature Set Detail Summary

Foundational services to be installed, configured and setup as part of the platform feature set.

Platform  
Development

**· Create ELK Stack for Log Aggregation and Analytics – Centralized log aggregation to be leveraged for structured data.**

**· Configure Elastic Filebeat on Jenkins Server to Push Logs to ELK – Installation and configuration of monitoring Jenkin Server logs, collecting the log events and ingestion to ELK stack for indexing.**

**· Configure Jenkins Statistics Analyzer to Forward to ELK – Enable Jenkins plugin to calculate build metrics related to standard deviation of build times, mean time to failure, mean time to recovery – to be ingested to ELK stack for indexing.**

**· Configure Kibana for Initial Dashboards and Alerts – Configure and setup Kibana dashboards for collection of data point visualization.**

**· Create Prometheus and Grafana Stack for Metrics Data – Setup and configure Promethues and Grafana stack for metrics collection data for Jira / Git and display in time-series presentation.**

**· Configure Jenkins Prometheus Plugin to Ship Build Metrics to Prometheus – Insallation and configure Prometheus plugin to expose endpoint (Jenkins) with metrics to be scraped.**

**· Configure Grafana with Initial Dashboards and Alerts – Setup of Grafana for time-series data for infrastructure and application analytics.**

**· Configure Application Nodes to send Application Logs to ELK – Configuration of application nodes to pull application node logs to ELK stack.**

**· Configure Application Nodes to send Health and Performance Metrics to Prometheus – Configuration of application nodes to pull metrics and send to Prometheus.**

<p>Tool chain selection and built-in accelerators constructed with structured / clean data flows to be available as open-source download.</p> <ul style="list-style-type: none"> <li>· <b>Define DevOps Cluster From Catalog of Supported Tools – Rest Service provide selection of tool based on service catalog to be constructed as desired toolchain. Tool chain will be constructed based on container / product versions available on dockerhub.</b></li> <li>· <b>Service to Automatically Provision Defined Cluster In Cloud (AWS) with One Click – Rest service to construct the Ansible / CloudFormation scripts to be executed on Cloud instance to build out foundational services (EC2, Security Groups, Route53, Docker Engine, Docker Network, Swarm) on target environment.</b></li> <li>· <b>Service to Automatically Provision Defined Cluster in Private Cloud - Rest service to construct the Ansible scripts to be executed on private Cloud instance to build out foundational services (Docker Engine, Docker Network, Swarm) on target environment.</b></li> <li>· <b>Service to Configure Application Links for Seamless Integration Between Tools – Rest Service to link tool set across the tool chain.</b></li> <li>· <b>Service to Inherit Jira Workflows, Jenkins Pipeline Definitions – Rest service to download workflows / global libraries for generating catalog of services that can be selected for customized delivery model.</b></li> <li>· <b>Configure Tools Monitoring and Metrics Gathering – Rest service to initialize health checks, monitoring and metric gathering of infrastructure / container components.</b></li> <li>· <b>Service to Configure Elastic Containerized Build Agents – Pre-defined build definition templates to construct on-demand build slaves optimizing resource utilization and consistent execution model.</b></li> <li>· <b>Service to Support Containerized Application Deployments – Analyzer to determine and construct DockerFile for container-based application types to generate and manage Docker images.</b></li> <li>· <b>Service to Enable Automated Version Management – UI and backend service to parse POM.xml to determine dependency management, apply hands-free artifact version updates based on adopted branching model.</b></li> </ul>	<p>Integrated Tool Chain Framework</p>
<p>Analytics feature set is focused on building the collectors for the top 5 tools that make up the tool chain. In the initial phase – we are focused on Git, Jenkins, Jira, SonarQube and Artifactory. The below data points will be collected and analyzed to be displayed in the UI/UX.</p> <ul style="list-style-type: none"> <li>· <b>Analytics on GIT – Branching, Pull Requests, Merges, Lead Times between Approvals, File / Line Changes &amp; Frequency, Common Errors Trends</b></li> <li>· <b>Analytics on Jenkins – Utilizing JenkinsFile, Stage Definitions / Standardization, Inheriting Workflows, CodeCoverage, Common Error Trends, Build Duration, Queue Times</b></li> <li>· <b>Analytics on SonarQube – Technical Debt, Code Smells, Tech Stack, Sloc Counts, Coverage, Change Frequency, Quality Gates (Pass / Fail Rates), Common Error Trends</b></li> <li>· <b>Analytics on Arifact Repositories – Vulnerability Metrics, Frequency of Change, Size Increase, Compliance Checks Reference to Aging Artifacts, 3rd Party Artifact Versions Frequency, Release / Snapshot Artifact</b></li> <li>· <b>Analytics on JIRA – Sprint Duration, Change In Story Estimates, Spill Over Stories, Number of Bugs, Code Quality Trends (#of Bugs, Issue Aging, foundInPhase), Scope Creep, WorkFlow Schemas, Utilization of Jira components, Duration between states, anti-patterns in workflows, FixVersion utilization, Field Utilization</b></li> </ul>	<p>Analytics Framework</p>

The feature set for the presentation layer is focused on a responsive web page – leveraging Node.js. Key features deliverables are focused on constructing the tool chain, deployment of tool chain to public/private cloud, accelerators configuration, identify / access management portal, rule engine configuration and real time analytics for 3 focus areas (time to market, engineering quality, and developer productivity)

UI / UX

- **UI Configuring Tool Chain Template – UI for tool selection and product versions to be constructed**
- **UI Deploy To Cloud – UI flow for providing end-points and credentials to configure and deploy to target environment.**
- **UI Artifact Dashboard – UI dashboard to configure and update artifact versions and apply changes per project/workspace**
- **UI Analytics Dashboard – UI view of 3 flows (product delivery & quality, engineering productivity, and operational / system metrics)**
- **UI Access Management – UI to manage user / group privilege to access portal and privileged tasks**
- **UI Rule Configuration Dashboard – UI to configure rules to filter resultset and render output to Analytics Dashboard**

#### Engineering Team Structure

The goal is focused on building a scrum team of 10 (co-located across US, Uruguay, and India) – initially can be run by Dean for a few months then we can hand-off to potential new hire (Eyo) who we would like to bring on board. Ideally – we'd like to get engineerings with mid-level experience who are highly motivated and can pick up new concepts / technologies and work against aggressive timelines. Myself and Dean will still be involved throughout the product lifecycle – playing the role of the technical product owners.

#### (2) UI Developers

**Skill Set – Strong Javascript skills including ES6+, experience with React+Redux**

**Years of Experience – 5+ years UI development experience**

**Tech Stack Utilized – ReactJS**

#### (3) DevOps Engineers

**Skill Set – Experienced with administering and configuring DevOps tools (JIRA, Bitbucket, Jenkins, Sonarqube, OpenShift, Nexus, Artifactory, etc), strong bash scripting skills, strong experience with Ansible, experience with Python, experience with working with different DevOps tools REST APIs, experience with Docker, experience with Cloud (AWS, GCP, and/or Azure), experience with ELK and Prometheus/Grafana.**

**Years of Experience – 5+ years DevOps experience**

**Tech Stack Utilized – Ansible, Python, various DevOps tools, Docker, OpenShift, AWS**

#### (4) Back End Developers

**Skill Set – Strong Python skills, experience with creating REST APIs, Machine Learning, Serverless development experience preferred**

**Years of Experience – 5+ years Python development experience**

**Tech Stack Utilized – Python, Serverless, REST API**

#### Product Delivery Timeline

The product feature cycle will be built to support monthly releases (2 week sprints) – driving incremental feature delivery available for review/feedback. We would ramp up the team one month prior to get them setup and trained on the product roadmap, development practices and incorporate them into the scrum team.

### **Month 1**

**Release Summary – Deliver the building blocks for the Integrated Tool Chain Framework, creation of UI for Configuration Tool Chain and Deploying Template to Cloud Type. End of month – we should have a portal to enable user to construct their tool chain and provision cluster on AWS cloud.**

**Resource Count - 1 UI, 1 DevOp, 2 BackEnd**

- **Define DevOps Cluster From Catalog of Supported Tools**
- **UI Configuring Tool Chain Template**
- **Service to Automatically Provision Defined Cluster In Cloud (AWS) with One Click**
- **UI Deploy To Cloud**
- **Service to Configure Application Links for Seamless Integration Between**

### **Month 2**

**Release Summary - Develop Accelerators for Integrated Framework to allow for structured practices and data set outputs, creation of Analytics for Source Code Management. End of month – we would have completed the plumbing for log aggregation and metrics gathering for GIT activities.**

**Resource Count - 1 UI, 2 DevOp, 3 BackEnd**

- **Service to Automatically Provision Defined Cluster in Private Cloud**
- **Service to Inherit Jira Workflows, Jenkins Pipeline Definitions**
- **Configure Tools Monitoring and Metrics Gathering**
- **Service to Configure Elastic Containerized Build Agents**
- **Service to Enable Automated Version Management**
- **Create ELK Stack for Log Aggregation and Analytics**
- **Create Prometheus and Grafana Stack for Metrics Data**
- **Analytics on GIT**

### **Month 3**

**Release Summary – Data Collector integration, data analytics capabilities for Jira / Jenkins and ingestion of data to ELK / Prometheus. At this point – we would be able to generate data points from Git, Jenkins, and Jira and publish the analytics to Grafana / Kibana.**

**Resource Count - 2 UI, 3 DevOp, 4 BackEnd**

- **Configure Elastic Filebeat on Jenkins Server to Push Logs to ELK**
- **Configure Jenkins Statistics Analyzer to Forward to ELK**

- *Configure Jenkins Prometheus Plugin to Ship Build Metrics to Prometheus*
- *Configure Kibana for Initial Dashboards and Alerts*
- *Configure Grafana with Initial Dashboards and Alerts*
- *Analytics on JIRA*
- *Analytics on Jenkins*

#### **Month 4**

***Release Summary – Log aggregation of Health Checks on Application / System, Complete Analytics for SonarQube and complete access management of UI portal for normal / privileged users.***

***Resource Count - 2 UI, 3 DevOp, 4 BackEnd***

- *Configure Application Nodes to send Application Logs to ELK*
- *Configure Application Nodes to send Health and Performance Metrics to Prometheus*
- *UI Access Management*
- *Service to Support Containerized Application Deployments*
- *Analytics on SonarQube*

#### **Month 5**

***Release Summary – Complete Analytics for Artifact Repositories, development of UI/UX for analytics dashboard, artifact management dashboard, and self-service rule configuration dashboard. This would conclude Phase 1 – with a comprehensive UI portal to manage onboarding, configuration and real-time metrics / log analysis based on the 3 focus areas.***

***Resource Count - 2 UI, 3 DevOp, 4 BackEnd***

- *Analytics on Artifact Repositories*
- *UI Analytics Dashboard*
- *UI Artifact Dashboard*
- *UI Rule Configuration Dashboard*