# SDLC Procedure & Continuous Engineering Practice

AWACS Development - Pune • 26.02.2021

# Overview

- Programming Languages
- Software Packaging
- SCM Tool  & Code Branching Policy
- Source Repositories
- Production Shakeout
- Code Quality & Coverage
- Issue Tracking
- API Implementation
- API Microservices (List)
- API Documentation
- Microservice Internal communication Methodology
- Authentication
- API Demonstration
- Performance Testing
- Library Packages
- API Integration
- API Secret keys
- Cloud Platform
- CSP Security Complience
- SSL Practice

# Programming Languages

## Java

- Standard spring cloud 2.3.x based microservice implementation using Java

## Go Lang

- Search Engine Optimization practices follows with search service using bleeding edge technology language

# Software Packaging

## Maven

- All java sub-modules are being packages by using java maven and registry is used as githubs package manager. A custom settings need to apply for all project M2_HOME to work with submodules.

## Docker Images & Docker Compose

- All software services are packages using docker images and registry is used as google cloud platform google cloud registry (GCR)

# SCM Tool & Code Branching Policy

## Git

- Standard git versioning tool is used as source code management tool.

## Branching Policy

- Standard main / master and feature / bugfixes policy refers, to work with code merge. Any third party contributor need to raise PR with owner of code in order to accept his code added in to mainstream code.

# Source Repository

https://gitlab.com/awacsdev

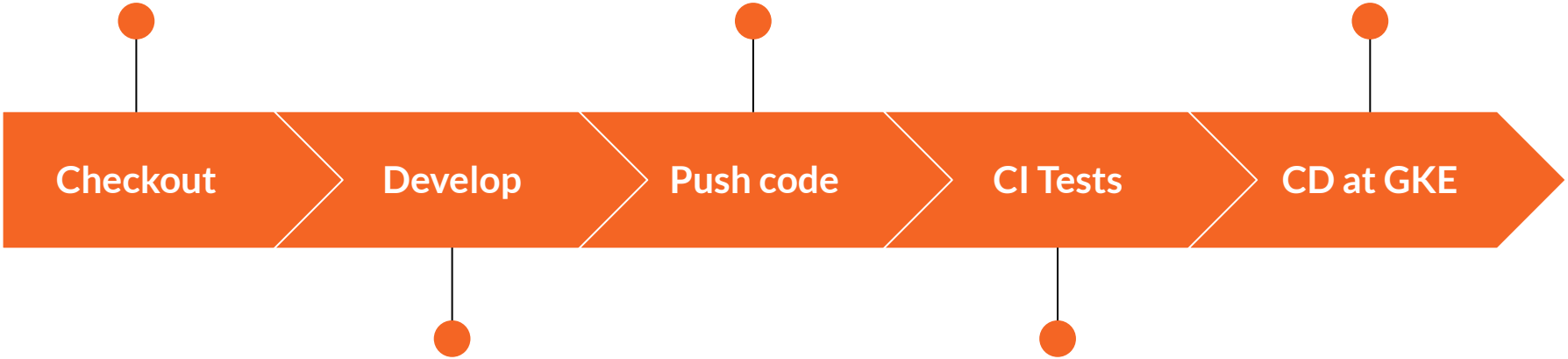
http://github.com/brkelkar

https://github.com/girishaiocdawacs/java-workspace.git
https://gitlab.com/thetechclan/

Standard checkout instructions

Uses github / gitlab source repository

After a series of tests, image, GKE cluster inside GCP prepared for kube patch




Developer contributes to feature request or bugfixes

Build & Packages Docker images and publish at gcr.io/awacs-dev/

Work flow

# Code Quality & Coverage

1. Sonar Cloud

   https://sonarcloud.io/organizations/girishaiocdawacs/projects

2. Git Guardian - Email alert in case any sensitive keys / secrets accidentally checks-in

# Production Shakeout

1. Uses github actions to shakeout periodic tests on running production instance.
2. It uses standard perf tests / sanity tests in a scheduled github workflows and raises an email in case there is a shakeout failure.

# Issue Tracking

1. Atlassian Jira

   https://awacs.atlassian.net/jira/projects

2. Atlassian Trello (scrum)

   https://trello.com/b/QS5TBYu9/smartpharmacy

   https://trello.com/b/K0EXEtvY/awacs-task-tracking

# API Documentation

1. Confluence Homepage
   https://awacs.atlassian.net/wiki/spaces/~165114925/pages

2. API Swaggers
   https://qa.awacscloud.tech/otpservice/swagger-ui/index.html?url=/otpservice/v3/api-docs

   http://smartapi.aiocdawacs.com/swagger/index.html

3. GraphQL Playgrounds

   http://app.awacsdev.com/search

   https://qa.awacscloud.tech/authserver/graphiql

# API Demonstration

1. Postman Newman collection ( Tool to send POST requests to server)
2. Newman Github actions

# Performance Testing

Following methodologies tried to approach API performance tests

- Apache Jmeter (Currently using)
- Blazemeter
- Apache Bench
- Hasura (hasura.io) - Regression
- Go Perf tool

# API Implementation

1. REST
2. GraphQL

# API Microservices

1. Search Service
2. Sync upload service
3. Auth Service
4. OTP  Service
5. Recaptcha service
6. Email service
7. Config service
8. Google Cloud Functions

# Microservice Internal communication Protocol

1. GRPC (Google Proto buf) for secure accessing APIs

2. Http - Rest for query internal functionality (e.g. send SMS)

# API Authentication

1. Oauth2 Implementation

- grant_type - client_credentials

  (for rest services access)

- grant_type - password
  (for user access)

# Library Packages

1. Maven submodules jar published here

   “awacs-cloud-commons”

   https://github.com/girishaiocdawacs?tab=packages

# API Integration

1. SMPP SMS - Text Nation SMSC
2. SMTP Email - Google Mail
3. SMTP  Email - Sendgrid (Planning)

# API Config & Secret Keys

1. Spring cloud & Google Storage is been used to park sensitive keys and private configs for the production APIs

# Cloud Platform

Google Cloud Platform (GKE Cluster) is our preferred choice of deployment and kubernetes embedded engine for deployments giving advantages of GCP’s stackdriver & managed services usage.

# Hostings & Compliances

1. Uses Google's nameservers from a1-5 regions

2. Deployments are Content Security Policy (CSP) Security Complience

   https://securityheaders.com/?q=qa.awacscloud.tech&followRedirects=on

# SSL Encryption

Supports TLSv1.3 and backward browsers compatibility

Sample Report - https://www.ssllabs.com/ssltest/analyze.html?viaform=on&d=qa.awacscloud.tech