Problem Statement Solution

1. Introduction

# Consumed API given in the problem statement and display the result as mentioned.

1. Deployment Solution

# I tried keeping all information as mentioned in problem statements. Taking two VMs to deploy UI and App component to make it load balanced. Below example is very high level of flow from App to the services and how request will be delegated to hosted zone.

# My App is the application (product UI) which will be used by the end user, once application sends request to any service then it will be via internet and accessible through https protocol. Initially request will be delegated to the DMZ zone and then it will be delegated to trusted zone and all services will be deployed in trusted zone.

# 

# As illustrated in below architecture diagram, we can divide our services based on group like platform service (basic service which must be executed) and business services (business micro services) and UI components. Below example represents primary environment and secondary one will be replica of primary

# **Component Group-1** – All basic services related to platform/framework (this could be company specific services as well) **Component Group-2** – All business services related to business functionalities **Component Group-3** – UI Services, this could be app or any API specific documentation (optional) Similarly another VM also will be loaded as primary and these two VMs will be loaded balance though any hardware load balancer like F5 LB.

Graphical user interface, application

Description automatically generated

A picture containing timeline

Description automatically generated

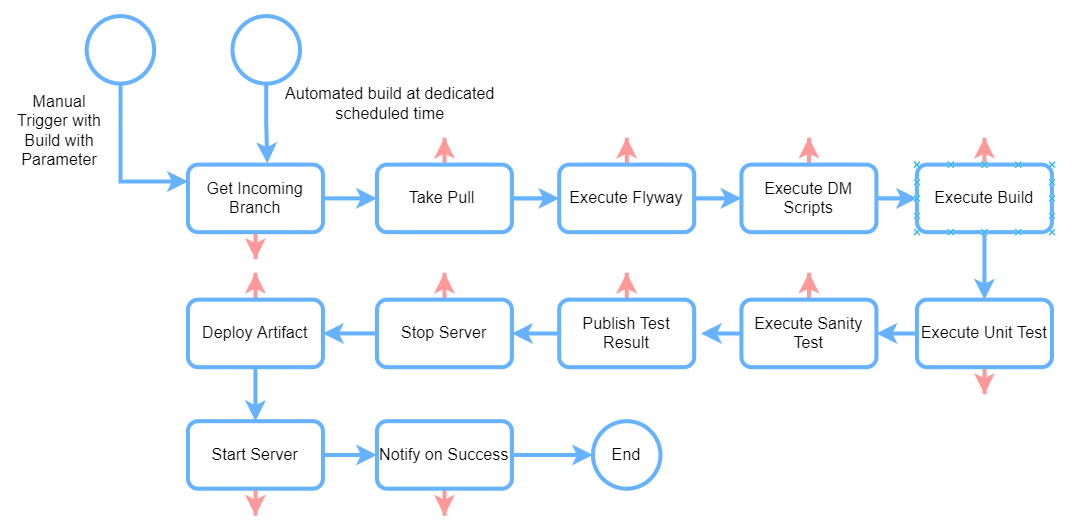
# Above architecture diagram is high level flow of API/services from the App, as illustrated any service call will be first generated from the app and app could be accessed via a browser and since API is access over https then complete content will be encrypted and send. At premises, https will be loaded as soon it reached to the load balancer. App is going to call an API with load balancer (could be F5 or any other). This LB will have configuration for API gateway and Juul can be implementation of API gateway. Jull will have all configurations and service URL which is deployed in Trusted zone.

# Direct service URL will not be exposed to the client, means App cannot call direct service URL instead it will be through load balancer and then will be delegated via gateway.

# As mentioned in above diagram, once requested reaches to Jull, it will identify SD and through SB request will be delegated to the product services and these services are deployed in two difference VMs

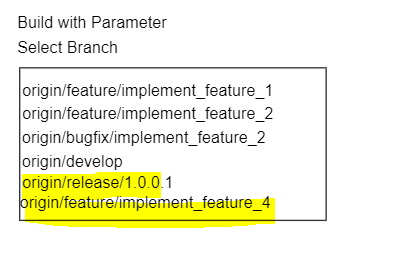
1. CICD Solution

# CICD stands for continuous integration and continuous delivery/deployment. This is nothing but automated process for getting code build from centralized server, verify testing, deploy in server, execute automation test, and notify users. This can be extended to deploy artifact in production environment. Below illustration is high level steps/process CICD take to deploy artifacts in production environment.



As per the problem statement, I tried keeping above diagram in very simple way. Once CICD pipeline is setup first point is to get code from centralized repository (it could be GIT) and take a pull for given branch. Depending on requirement, we my need to execute few DDL/DML, to do that we can setup Flyway job as next process/job which can take care of DDL and DML execution, once it’s done then build process will start. Once build is success there will be test suits which must be executed and if test results are within threshold (we can set the same), we can publish the test result and start deployment. To do this, first Jenkin to execute script to stop server, deploy artifact in server and start the same. Since we have clustered environment, deployment can be in rolling process i.e., stop one server deploy the same and start the server and once server is up then go to another server to do the same action. If any issues during any steps then process will be rollbacked and notification will be sent to the DevOps team. If any DDL/DML or Flyway is executed, then flyway should execute prior version of flyway script so that DDL and DML can be rollbacked.

# In Jenkin we can setup dedicated build based on environment like Dev/QA/UAT/PROD. If its dev environment scheduler Job will pick develop branch and proceed with the pipeline has been setup and similar for another environment as well. We can have another option where input branch will be passed by the user who will trigger the build, in this case use should be able to select branch from the repository like, if for QA team there will be an option to deploy any branch in QA environment, it could be develop/feature/release/bugfix depending on requirement and urgency. Belo snippet is just for a reference how user can select branch in Jenkin and trigger the build.



1. Docker file and Documentation

# A docker file has been created for this application and while build it will generate jar file under target folder. Go to the application root and open power shell and hit below command

# Create docker image: docker build -t wapp .

# Check if image is created: docker images

# Run docker image: docker run -t -p9090:8080 wapp

# Graphical user interface, text Description automatically generated with medium confidence

# Once docker container has started you can hit swagger command below.

# Graphical user interface, text, application Description automatically generated

1. Assumption

# I explained deployment and CICD as per my understanding. I have deployed many products in Airline Cargo Industry. These products are deployed in premise whereas it is being access via cloud services. This architecture holds good for microservice deployment as I tried taking almost component used in M/S architecture. I assume that if user is running this application on any machine, then below mention environment must be available. Please look at readme.txt file under source code to get more on tech stack.

# Java: version8

# Maven: 3.6.\*

# 