Application containerization blueprint report

For

Passport verification API

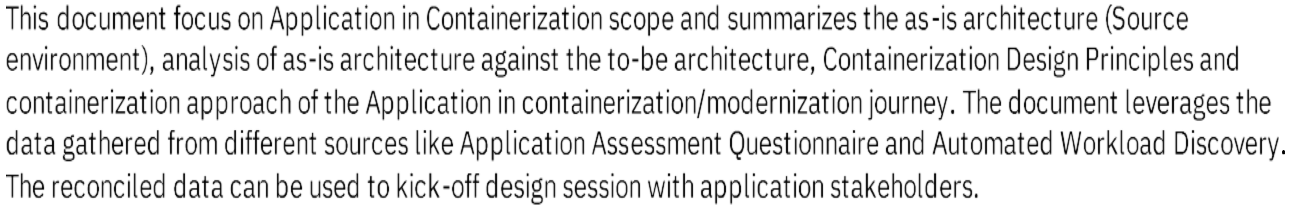
(G123OPABC)

VERSION CONTROL:-

Version control for blueprint reviews.

VERSION		REVISION	REVIEWER
V0.1		Initial Draft	
V.2		Add AWS account and clarify upstream/downstream	
v.3		Added Section 3.3.3 Technology Stack 4.2.2.3 Component Interactions Updated the links in section 9 - Tokens / Secrets	
v.4		Added AS-IS and To-BE architecture	

1. INTRODUCTION.



This document focus on Application in Containerization scope and summarizes the as-is architecture (Source environment), analysis of as-is architecture against the to-be architecture, Containerization Design Principles and containerization approach of the Application in containerization/modernization journey. The document leverages the data gathered from different sources like Application Assessment Questionnaire and Automated Workload Discovery. The reconciled data can be used to kick-off design session with application stakeholders.

2. SUMMARY

2.1 APPLICATION PROFILE

Application profile contains the details of planned/targeted for containerization.

Application name	Passport verification API
System code	AZ12345Z
App code	Ad12tyy6
Repo	https://gitlab.com/upskillondevops.com branch: feature_devops
Aws account details	Dev: pass_verify_dev 342578981 Si: pass_verify_si 784645898 Prod: pass_verify_prod 987645678
ROSA server details	Dev: https: dbr.htyet.dev.openshift.com Si: https: dbr.htyet.si.openshift.com Prod: https: dbr.htyet.prod.openshift.com
Application manager	Jamesthomas
Cost centre	App12345
App description	This application will serves as a API for main application.

This application profile containe tags which is also used as resources grouping for you application in aws/rosa.

3. Server environments:

3.1 existing server details.(non-containerization)

this source server container details like server id, server name, credentials.

App server environment(dev/si,prod) ,role, os, softwares installed etc.

Web server

Database server

3.2 target server details (containerization)

Namespace	FQDN		
can-acc-doc-api-prd	https://myserver.dns.com	Production	/redhat-openjdk-18/ openjdk18-openshift/images/
		DEV	/redhat-openjdk-18/ openjdk18-openshift/images/
		System Integration	/redhat-openjdk-18/ openjdk18-openshift/images/

3.3 Upstream/Downstream Dependencies

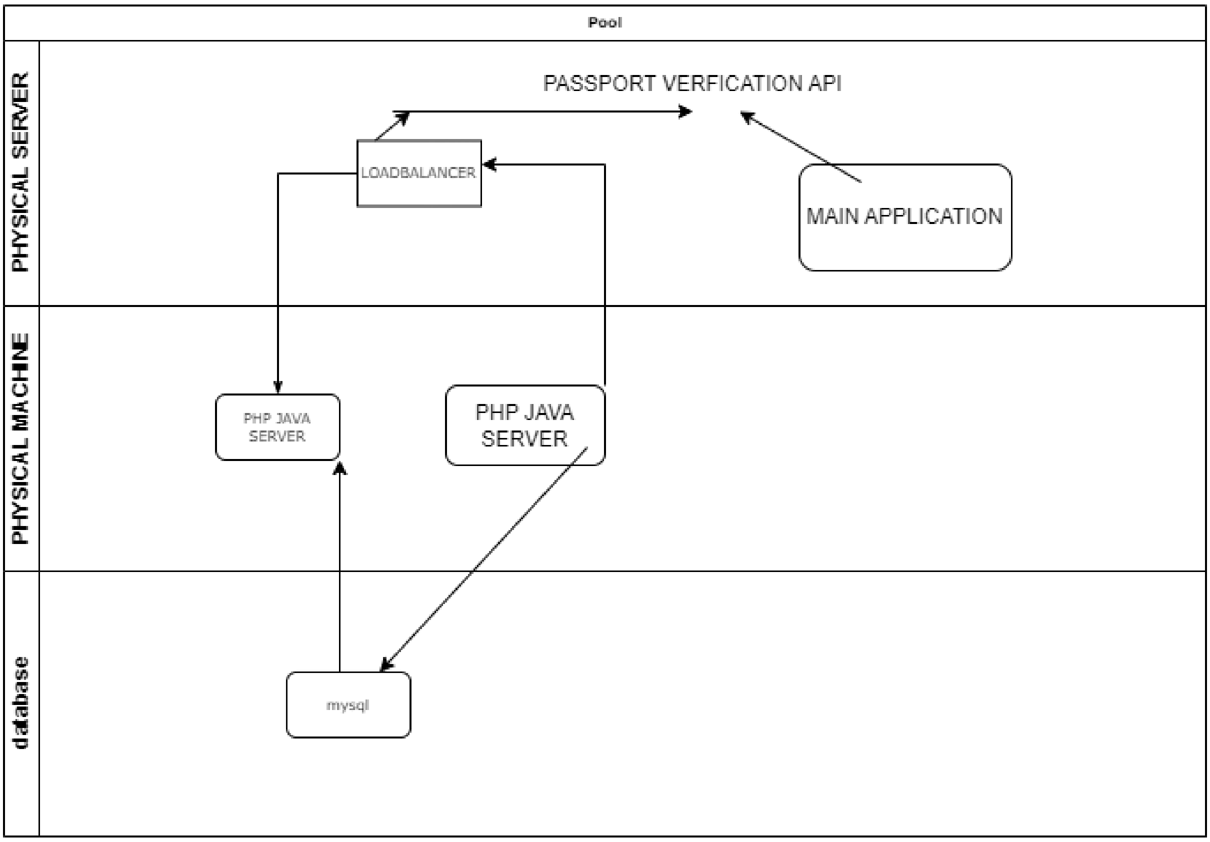
The list needs to include details for all environments (SI, Dev, and Prd).

Upstream (Who/what does this application call or depend upon)

PASSPORT CREATE REQUEST API

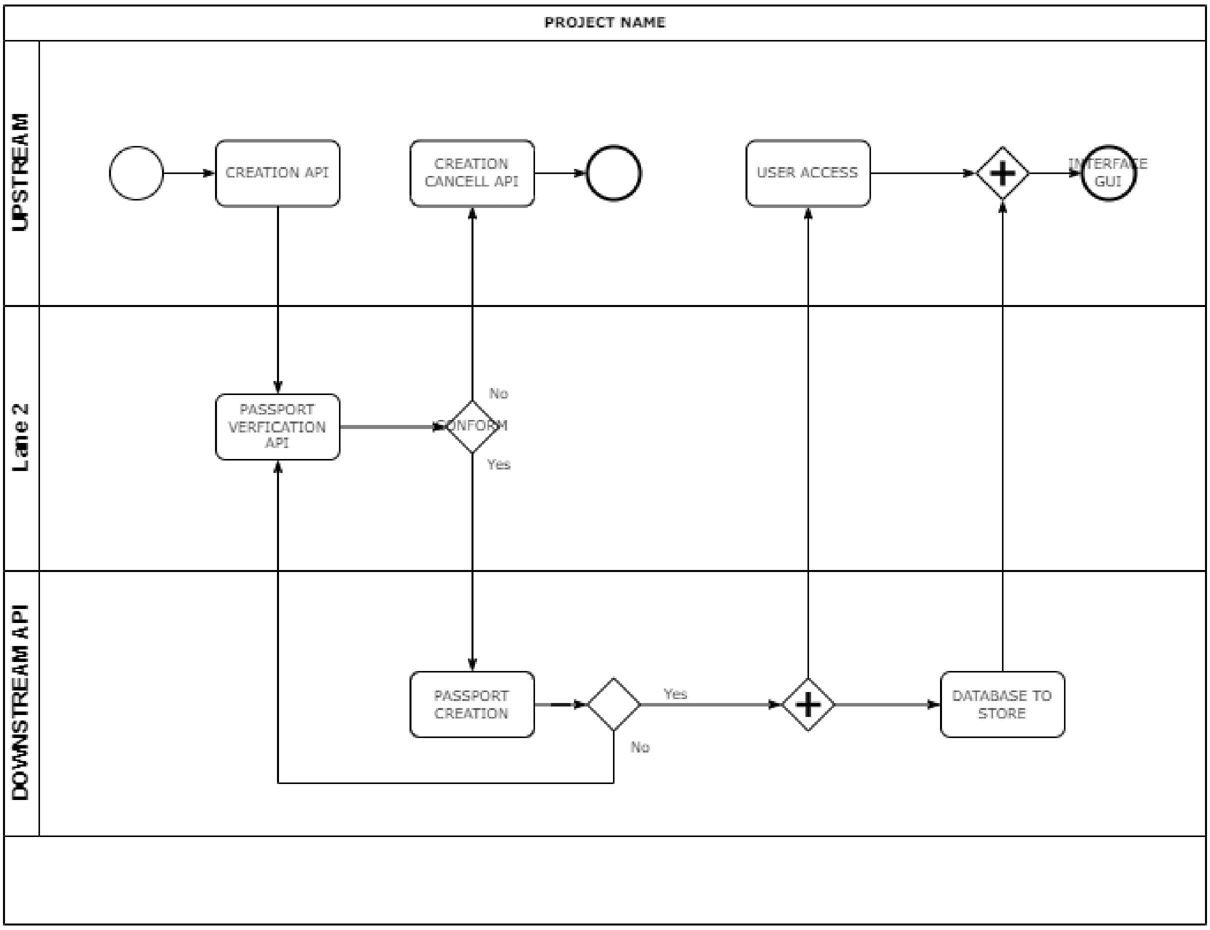
Ticket V3	https/ 37526	Yes	Yes	On-Prem	Planned for wave 8-15	https:webservice.com
	https/ 31063	Yes	Yes	On-Prem	Planned for wave 9-13	

4. CURRENT APPLICATION ARCHITETURE



PHYSICAL ARCHITECTURE. This is physical server architecture was setup in locations.

4.1 PASSPORT VERFICATION API INTERACTION UPSTREAM/DOWNSTREAM FLOW



5. Source server utilization
   5.1 Cpu utilization

Passport api	Production	21.03	21.03	21.03

   5.2 memory utilization

Passport api	Production	2.75	-	-	-
	Production	2.68	-	-	-

   5.3 disk utilization

can-acc-doc-api-prd	Production	2.75	-	-	-
fulfill-rule-ms-prd	Production	2.68	-	-	-

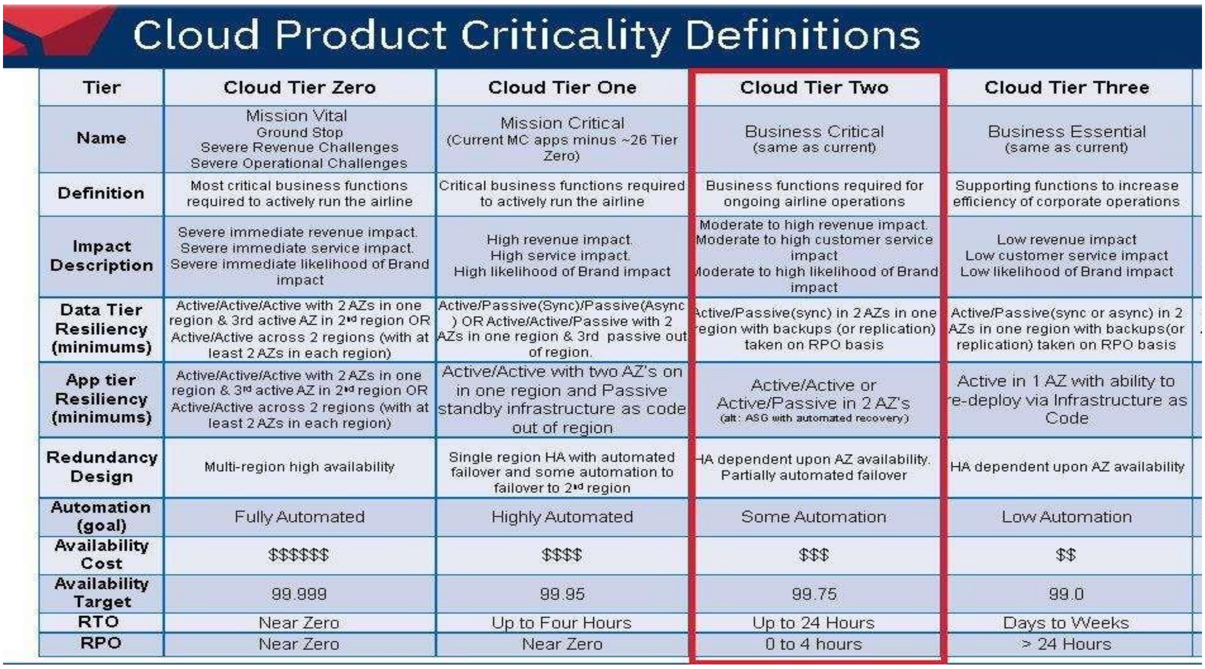
6. usage metrics
   6.1 application transition rate
   6.2 databse transition rate

7. source target analysis

containerization analysis

No of deployments	1 pod
Namespace	Namespace name
Technology analysis	Java version: 17 Stack: springboot Database: mysql Baseimage: linux Technologies: springboot, java, hibernate Server: tomcat.

8. Recommended containerization analysis.



Cloud Product Criticality Definitions

Tier	Cloud Tier Zero	Cloud Tier One	Cloud Tier Two	Cloud Tier Three
Name	Mission Vital Ground Stop Severe Revenue Challenges Severe Operational Challenges	Mission Critical (Current MC apps minus ~26 Tier Zero)	Business Critical (same as current)	Business Essential (same as current)
Definition	Most critical business functions required to actively run the airline	Critical business functions required to actively run the airline	Business functions required for ongoing airline operations	Supporting functions to increase efficiency of corporate operations
Impact Description	Severe immediate revenue impact. Severe immediate service impact. Severe immediate likelihood of Brand impact	High revenue impact. High service impact. High likelihood of Brand impact	Moderate to high revenue impact. Moderate to high customer service impact Moderate to high likelihood of Brand impact	Low revenue impact Low customer service impact Low likelihood of Brand impact
Data Tier Resiliency (minimums)	Active/Active/Active with 2 AZs in one region & 3rd active AZ in 2nd region OR Active/Active across 2 regions (with at least 2 AZs in each region)	Active/Passive(Sync)/Passive(Async) OR Active/Active/Passive with 2 AZs in one region & 3rd passive out of region.	Active/Passive(sync) in 2 AZs in one region with backups (or replication) taken on RPO basis	Active/Passive(sync or async) in 2 AZs in one region with backups(or replication) taken on RPO basis
App tier Resiliency (minimums)	Active/Active/Active with 2 AZs in one region & 3rd active AZ in 2nd region OR Active/Active across 2 regions (with at least 2 AZs in each region)	Active/Active with two AZ's on in one region and Passive standby infrastructure as code out of region	Active/Active or Active/Passive in 2 AZ's (alt: ASG with automated recovery)	Active in 1 AZ with ability to re-deploy via Infrastructure as Code
Redundancy Design	Multi-region high availability	Single region HA with automated failover and some automation to failover to 2nd region	HA dependent upon AZ availability. Partially automated failover	HA dependent upon AZ availability
Automation (goal)	Fully Automated	Highly Automated	Some Automation	Low Automation
Availability Cost	$$$$$$	$$$$	$$$	$$
Availability Target	99.999	99.95	99.75	99.0
RTO	Near Zero	Up to Four Hours	Up to 24 Hours	Days to Weeks
RPO	Near Zero	Near Zero	0 to 4 hours	> 24 Hours

Multi region, high availability, availability zone, rpo, rto guidelines.

9. Target cluster technology details:

Upskillondevops	Dns name	Production	us-east-1	Openjdk:17 -	ROSA	N/A	AZ1, AZ2 and AZ3

10. Target shared environment details

10.1 database backup

Prod App Server	Daily @ 0300	35 days	Snapshot stored in S3	Delete after 35 days
SI App Server	Daily @ 0300	7 days	Snapshot stored in S3	Delete after 7 days
Dev App Server	Daily @ 0300	7 days	Snapshot stored in S3	Delete after 7 days

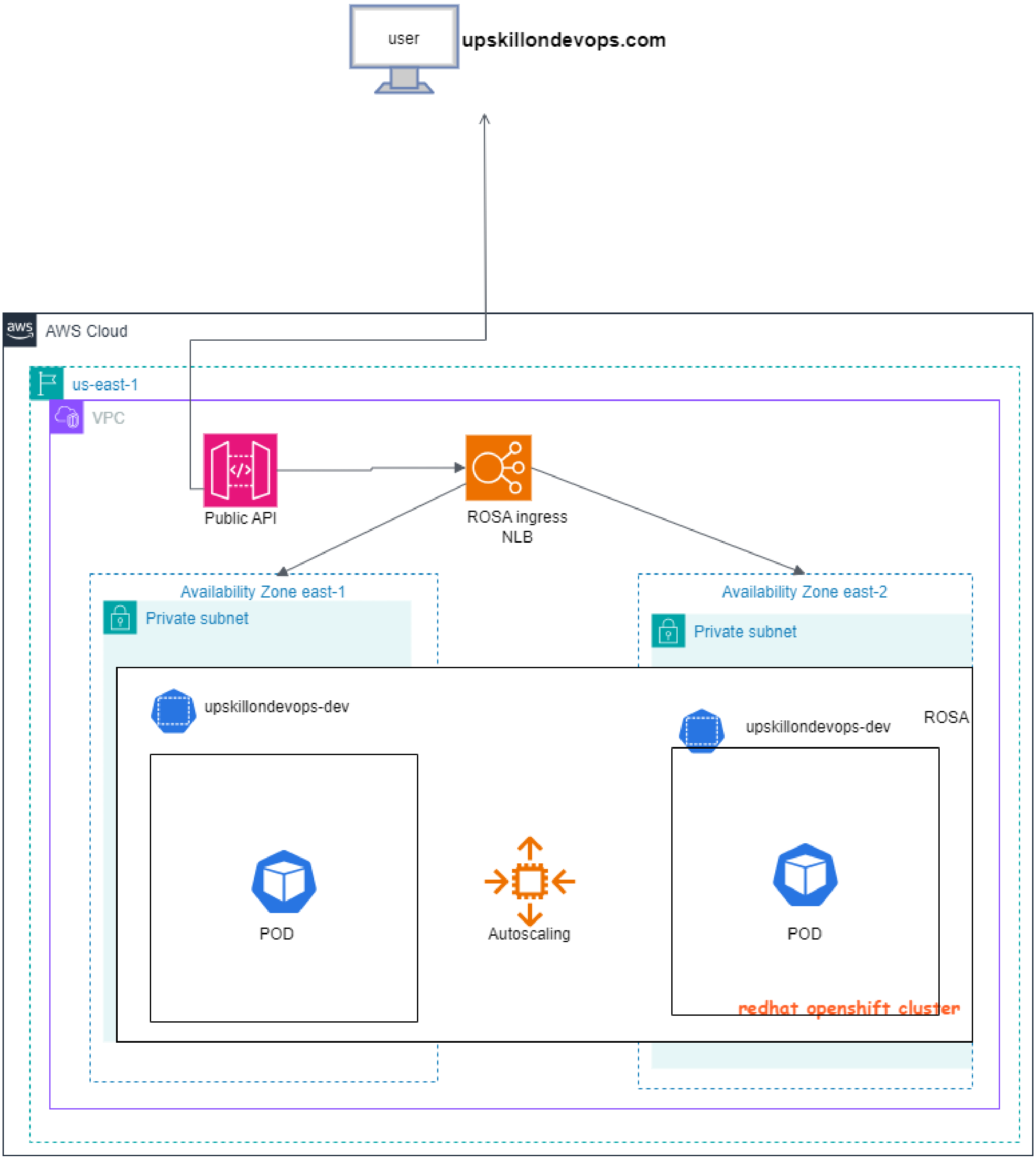
11. source and target details

DAL-SP-045	RHEL N-2 & Below	N-2	DAL-TP-032	Red Hat OpenShift on AWS - PaaS (ROSA)	N
DAL-SP-026	SpringBoot	2.1.x	DAL-TP-052	OpenJRE base image (Deployed as Spring boot Framework enabled API	Latest based on CCOE
DAL-SP-009	IBM MQ		DAL-TP-012	Amazon MQ	
DAL-SP-002	Oracle	11	DAL-TFP-019	AWS RDS (PostgreSQL)	N

12. database migration details.

Prd	Oracle	Amazon RDS(PostgreSQL)	Full load & CDC	AWS DMS, SCT and CFT
SI	Oracle	Amazon RDS(PostgreSQL)	Full load & CDC	AWS DMS, SCT and CFT

13. target application architecture.

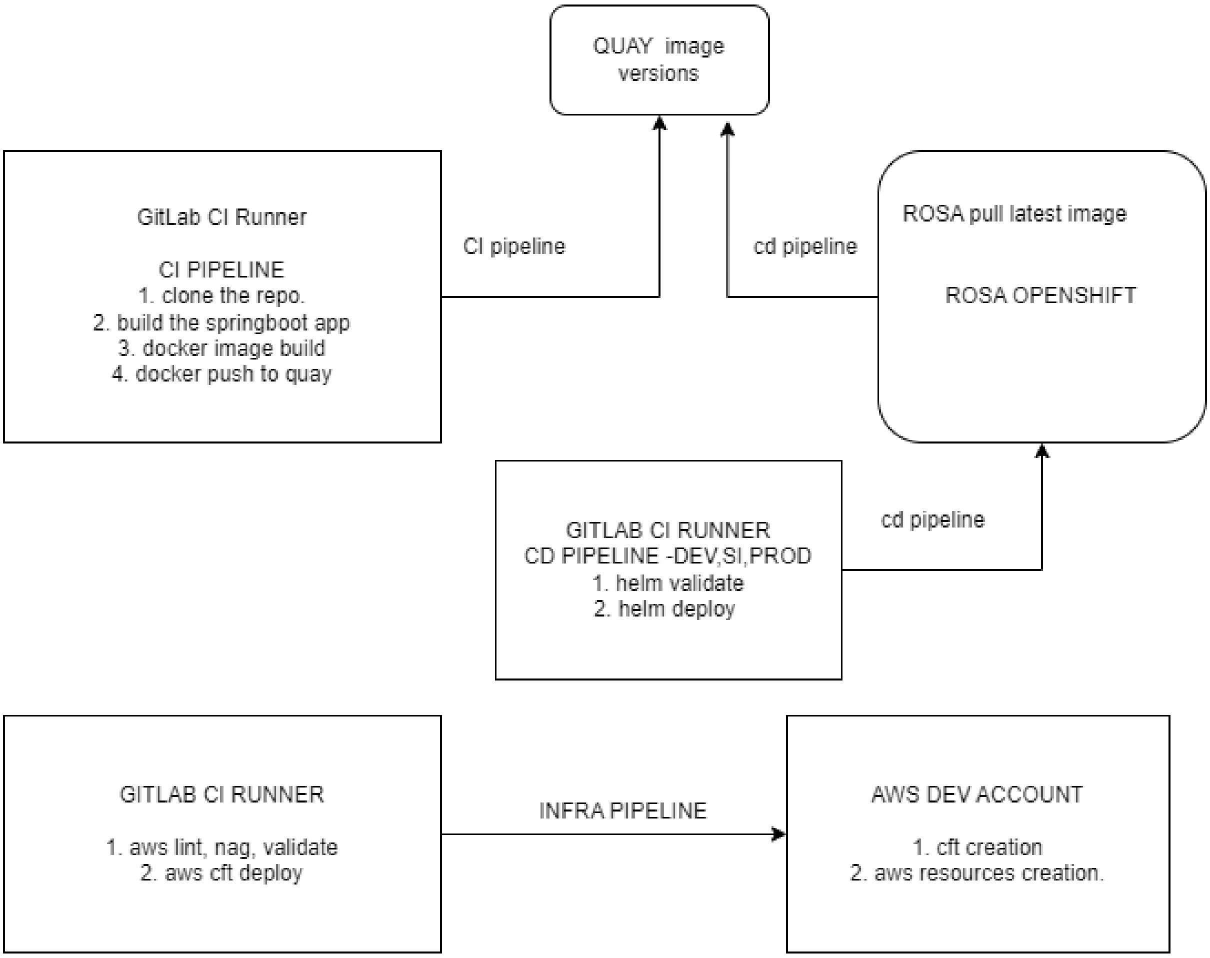


1. upskilldevops.com is main application and passport verification is API.
2. When passport api is hit by the client then route 53 resolves the right DNS (in this demo we are skipping route 53 we can custom the api entry in hosted zone)and request in route to the api gateway.
3. Api gateway is routing to the rosa ingress NLB (in this demo we are using rosa sanbox)
4. Rosa ingress NLB having VPC endpoints pointing to the rosa cluster pods ips to access the application.
5. Apigateway is montoring the logs.

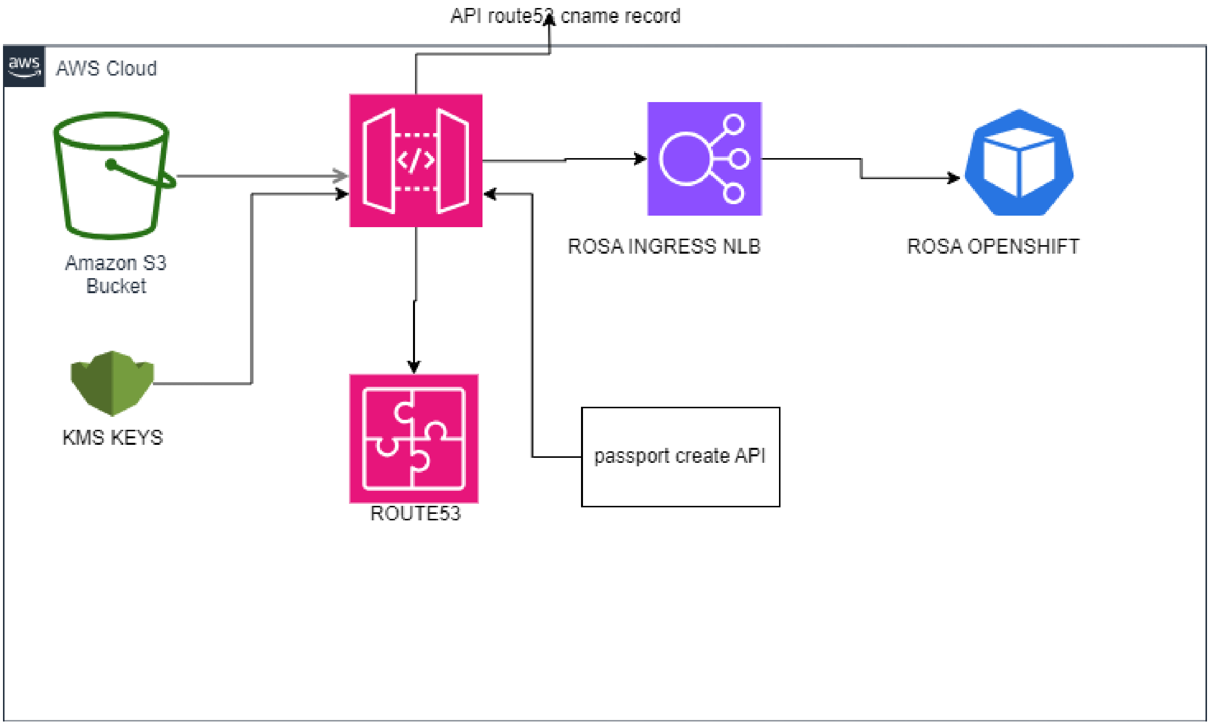
14. Incoming traffic in containerization

We need to integrate multiple api using entries in route53, and thenroute 53 resolved to this apigateway enpoint.

15. KEY ASSUMPTIONS:
    15.1 DEVOPS PIPELINE: GITLAB CI
    15.2 ROSA
    15.3 DEVOPS ARCHITETURE FOR DEPLOYMENT



    15.4 aws infra-architecture.



REVIEWS:

16. VULNERABLITY REPORTS SAST,DAST.
17. TAGS
18. SECRETS: like aws secrets, rosa secrets, quay secrets.