**Application containerization blueprint report**

**For**

**Passport verification API**

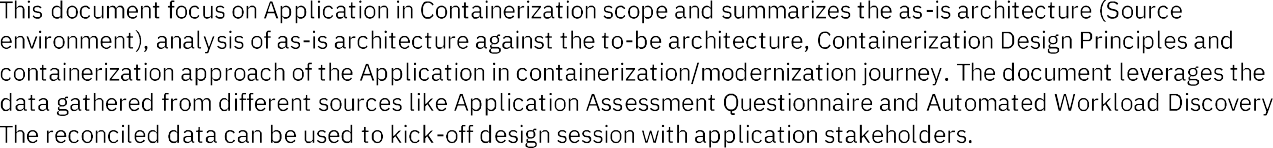
**(G123OPABC)**

VERSION CONTROL:-

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION |  | REVISION | REVIEWER |
|  |  |  |  |
|  |  | A black background with white text  Description automatically generated |  |
|  |  |  |  |
|  |  |  |  |

Version control for blueprint reviews.

1. **INTRODUCTION.**



1. **SUMMARY**
   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.

1. **Server environments:**
   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**

* 1. **target server details (containerization)**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | BASE IMAGE |
| A black background with white text  Description automatically generated | https://myserver.dns.com |  | A black background with white text  Description automatically generated |
|  |  | DEV | A black background with white text  Description automatically generated |
|  |  | A black background with white text  Description automatically generated | A black background with white text  Description automatically generated |
|  |  |  |  |

3.3 





PASSPORT CREATE REQUEST API

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | https:webservice.com |
|  |  |  |  |  |  |  |

1. CURRENT APPLICATION ARCHITETURE

A diagram of a computer application

Description automatically generated

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

* 1. PASSPORT VERFICATION API INTERACTION UPSTREAM/DOWNSTREAM FLOW

A diagram of a project

Description automatically generated

1. Source server utilization
   1. Cpu utilization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Passport api |  |  |  |  |

* 1. memory utilization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Passport api |  |  |  |  |  |
|  |  |  |  |  |  |

* 1. disk utilization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. usage metrics
   1. application transition rate
   2. databse transition rate
2. 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. |

1. Recommended containerization analysis.

A close-up of a product criticality definition

Description automatically generated

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

1. Target cluster technology details:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| Upskillondevops | Dns name |  |  | Openjdk:17 |  |  |  |

1. Target shared environment details
   1. database backup

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  | A black background with white text  Description automatically generated |
|  |  |  |  |  |
|  |  |  |  |  |

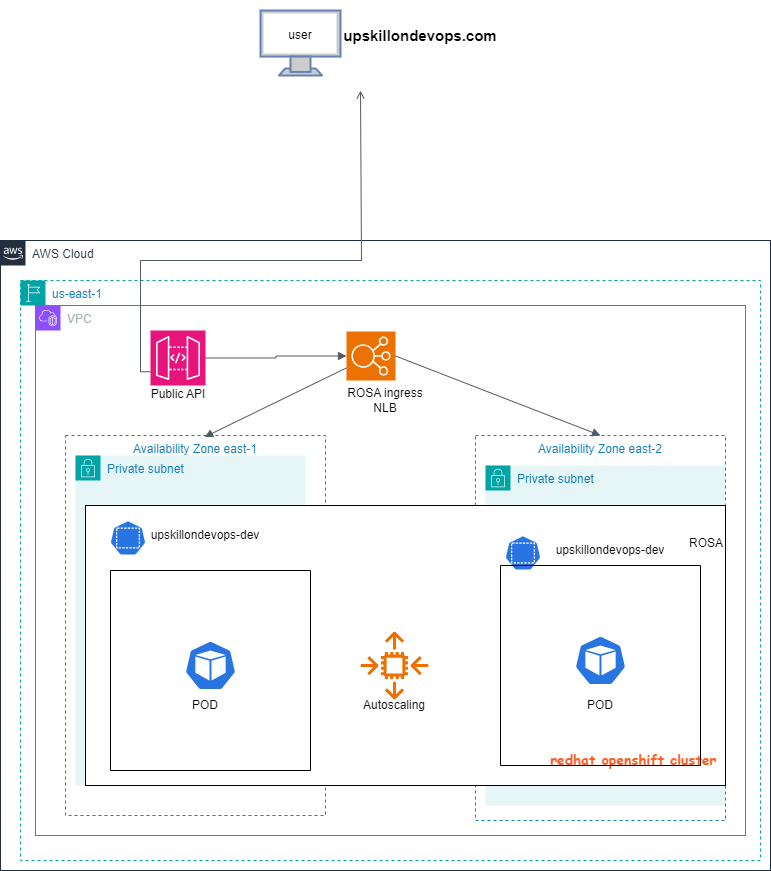
1. source and target details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  | A black background with white text  Description automatically generated |  |
|  |  |  |  | A black screen with white text  Description automatically generated | A black background with white text  Description automatically generated |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. database migration details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

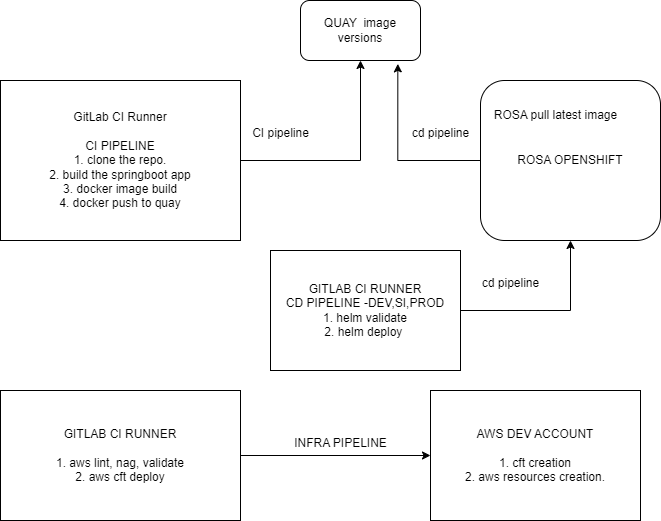
1. **target application architecture.**



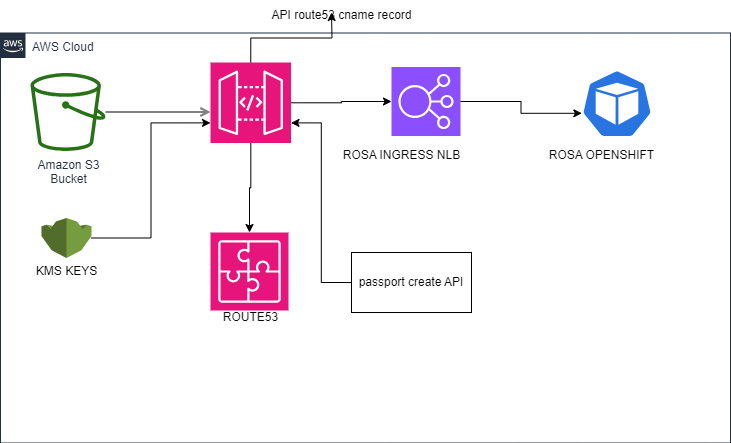
1. upskilldevops.com is main application and passport verification is API.
2. When passport api is hit by the client then route 53 resolves the right DNS (in this demo we are skipping route 53 we can custom the api entry in hosted zone)and request in route to the api gateway.
3. Api gateway is routing 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.
6. Incoming traffic in containerization

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

1. KEY ASSUMPTIONS:
   1. DEVOPS PIPELINE: GITLAB CI
   2. ROSA
   3. DEVOPS ARCHITETURE FOR DEPLOYMENT



* 1. aws infra-architecture.



REVIEWS:

1. VULNERABLITY REPORTS SAST,DAST.
2. TAGS
3. SECRETS: like aws secrets, rosa secrets, quay secrets.