**Application Architecture**

Service Registry

/employee/

Employee

Department

Spring cloud API Gateway



Dashboard

Config Server

/departments/

**Architecture : Spring boot services**

* Microservices – Employee & Department
* Both Microservices are connected from Spring cloud API gateway.
* Users can only be able to access through API gateway
* Historic dashboard will show services behaviors & resilience ratio
* All are connected to service registry
* Config server get configuration from GitHub repo

**Architecture : Docker**

* Docker file maven plugin to create docker images
  + API Gateway
  + Employee
  + Department
  + Dashboard
  + Config server
  + Service Registry
* Images pushed to Antifactory/Docker hub

**Architecture : Kubernetes**

* Deployment & Service components – internal services (not directly access from outside)
  + Employee
  + Department
  + Config server
  + Services are created as ***cluster IP***
* API Gateway – deployment & ***Load balancer*** Service (all requests are coming through)
* Dashboard – deployment & ***Node port*** service (Port will be enable for particular virtual machine)
* Service registry – Stateful set (Have to maintain the host information)

& ***Headless*** service (to create end points internally to expose the pod to the K8s cluster)

* Pod can be access : ***podname-{replica index}.{serviceName}.default.svc.cluster.local***
* All container images are added to the Kubernetes cluster.

**Architecture : AWS Services**

* Pre-requisites : install AWSCLI, kubectl, aws-iam-authenticator
* Role need to created to access K8s cluster
* Create VPC & Nodes using Terraform
* Create K8s cluster using AWSCLI
* Map notes to the Master

**Pre-requisites : Jenkin Pipeline**

* Install Git, Java, Maven, Docker & Jenkin
* Install Ansible (Playbook will do the deployment through Jenkin)
* Install Python
* Install K8s module (openshift) – pip3
* EKS Cluster

**Architecture : Jenkins**

* Integrate Jenkins shared library
* Required Plugins
  + AWS steps for pipeline
  + Maven
  + Kubernetes
  + Kubernetes continues deploy
  + Kubernetes credentials proiver
* Credentials
  + Docker
  + Kubernetes
  + AWS

**Architecture : Microservice to AWS EKS**

* Create a spring boot application
* Create Docker images from spring boot application using dockerfile
* Docker image pushed to Antifactory/Docker hub
* Ansible will be pull the Docker image from Antifactory/Docker hub using playbook
* Docker container added to the K8s clusters

**Architecture : EKS Monitoring**

* Elastic search
* Fluentd
* Kibana

**Pre-requisits**

* Install eksctl utility
* Configure aws account using awscli

**Configurations**

1. Create a K8s namespace for monitoring
2. Install Elastic search using Helm
3. Install Kibana using Helm
4. Install Fulentd & make sure that running as a demonset
5. Install dapr with enabling JSON-formatted logs
6. Deploying and viewing application logs
7. To search logs - Get kibana GUI using port forwarding

**Architecture : Argo CD**